

© 2008 Chad E. Peiper

A TEACHER'S DASHBOARD: MONITORING STUDENTS IN TABLET PC
CLASSROOM SETTINGS

BY

CHAD E. PEIPER

B.S., Case Western Reserve University, 1994

M.A., University of California at Santa Cruz, 1996

M.C.S., University of Illinois at Urbana-Champaign, 2000

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Doctoral Committee:

Associate Professor Samuel Kamin, Chair

Professor Leonard Pitt

Associate Professor Michael Twidale

Professor Tim Wentling

Associate Professor Cynthia Carter Ching, UC Davis

Abstract

Networked Tablet PCs have great potential for improving the learning environment in classrooms. By increasing the capacity for *communication* among participants, they can raise student engagement in, and teacher awareness of, the class; by allowing for a *detailed record* of the class to be made, they can facilitate educational research. This thesis presents several studies, spanning the period from fall, 2001, to spring, 2008, exploring these possibilities. An early study used primitive pen-enabled computers to allow sharing of teacher notes and student questions. In several studies, Tablet PCs were used simply to record data, either of the teacher's use of the Tablet PC or of the students' note-taking; we present a variety of analyses of those data. In the spring, 2007, semester, we experimented with a partially self-placed class structure made possible by the use of the Tablet PC. In our last study, several classes were conducted in a traditional manner, but with all participants equipped with Tablet PCs; the sole intervention was the provision of a "dashboard" on which the teacher could see the students' work. In addition to presenting results of student surveys, we discuss the design issues for such dashboards.

Dedication

This dissertation is dedicated to the memory of Professor Michael Faiman.

Acknowledgments

I have many people to thank and I apologize in advance to anyone reading this who cannot find their name.

First, to my mother, she worked three jobs so that we could have a place to live, food to eat, and clothes to wear. She boldly convinced my high school principal that I was “smart enough” to be in the pioneering “high-tech” program and that I should be placed in all of the advanced placement courses. During my circuitous journey it was my mother who supported me emotionally and inspired me to succeed. Didn’t you know? Mothers are *always* right, it was back in 1984 when mine said, ”...get the Apple IIe, it’s more ver-sa-tile!”

To the rest of my family for all of their love and support

Claudia Mruk may she reset in peace.

- Life just before coming to the University of Illinois at Urbana-Champaign
 - David Cope for teaching me my first programming language, LISP
 - Jim and Kathy Horner for all of their magnanimous support and for inspiring me to be successful
- During my transition from Musicology PhD student to Computer Science Lecturer
 - Musicology Professors John Hill, Tom Ward, and Nicholas Temperley who supported me and helped me to find my way to Computer Science (it’s on the “north side” of campus).
 - Professor Caroline Hayes for her generous time supporting my independent study of artificial intelligence
 - Scott Parker for sitting down with me in the summer of 1998 and helping me write my first C++ program.
 - Jason Zych for teaching me to teach
 - Barb Cicone for helping me transition into Computer Science (it was my dream for so long). Barb has been a constant source of support to our graduate students, instructors, and professors.
 - Professor Harandi for entrusting me to teach CS302 when no one else would, for allowing me to teach CS300 when none else could, and for giving me the opportunity to pursue my love of teaching.
- During the e-Fuzion era

- Patrick Bristow and David Pan
- Tim Wentling and his Knowledge & Learning Systems Group for generously giving of their time, energy, resources, and pedagogical expertise in supporting my research interests.
- Funding:
 - * The Academy for Excellence in Engineering Education (AE³) for funds to help us develop e-Fuzion
 - * Microsoft Research and Hewlett Packard for their generous grants
 - * Department of Computer Science for supporting my research
 - * Fujitsu Computer Systems for their generous gifts and support
- People:
 - * Professor Muroga for introducing me to Fujitsu and their executives George Wintner and Vic Goldblum of Fujitsu Computer Systems
 - * Kay Tomlin for providing countless years of support and for facilitating the creation of all the pilot studies presented in this thesis
 - * Chris Migotsky and Kathy Linger for helping me to earn the Advanced Graduate Teaching Certificate
- All my teaching assistants and course staff assigned to me over the years. Special thanks to Manish Agrawa, Karan Khanna, Zackary Runner, Bardia Sadri, Alex Wallace, Petre Tzvetkov, and David Warden.
- Administrative folks in the department who we all *know* really run the department. They always, without hesitation brought a smile to my face and encouraged me to push forward and succeed: Erna Amerman, Barbara Armstrong, Holly Bagwell, Trisha Benson, Barbara Cicone, Molly Flesner, Shirley Finke, Steve Herzog, Mary Beth Kelly, Rhonda McElroy, Lori Melchi, Anda Ohlsson, and Kay Tomlin
- Study participants who generously gave of their time and energy:
 - Instructors: Lawrence Angrave, Eugene Bild, Ioana Boca, Marco Caccamo, Kevin Chang, Larry Debrock, Jena Finch, Cinda Hereen, Craig Russell, Craig Zilles, and Jason Zych
 - The many students who enrolled in CS300/CS400, CS125, CS225 sections set up for our pilot studies
 - University High Laboratory student participants in Algebra I, Algebra II, Geometry, and Spanish
- Department of Computer Science:
 - Administrators: Mark Snir, Mehdi Harandi, Michael Heath, and Dennis Mickunas
 - Professors: Geneva Belford, Roy Campbell, Jeff Erickson, George Friedman, Guy Garnett, Mehdi Harandi, Ralph Johnson, Robin Kravets, Klara Nahrstedt, Lenny Pitt, Dan Roth, Lui Sha, and Marianne Winslett
 - TSG Staff: Drew MacGregor, Chuck Thompson, Darwin Miller, Teresa Schrug, Frank Penrose, Scott Cimarusti, and Barb Leisner
 - Rick Henderson, Rick Van Hook, and Patrick Patterson
- Personals:
 - Bill Roberts, Ruxandra Sevastru, and George Wintner
 - Ellick Chan, Mark Sammons, Andrew Wadsworth, and Tanya Y. Berger-Wolf
 - Dr. Francis C. Lee, M.D. and Veronica Lee, Sherban and Daniela Lupu, Ian Hobson and Laura Greene, Michael Becker of Fine Violins Ltd., and Rebecca and Robert Riley

- I would like to also thank the following people for going *out of their way* to make this goal a reality: Roy Campbell, Barbara Cicone, Boris Capitanu, Samuel Kamin, Kay Tomlin, David Warden, and Robert Warden
- Jason Zych who championed my interests and inspired me to succeed
- The members of my committee for all of their patience, sage advice, and never ending support: Cynthia Carter Ching, Lenny Pitt, Michael Twidale, Tim Wentling and Sam Kamin

My advisor Sam Kamin. Words can not express my heartfelt gratitude.

Table of Contents

List of Tables	xi
List of Figures	xii
Chapter 1 Introduction and Motivation	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Overview of Dissertation	5
1.4 Contributions	7
1.4.1 Assessing Tablet PCs impact on student learning	7
1.4.2 Analysis of student attitudes using Tablet PCs	7
1.4.3 Analysis of teacher behaviors in accommodating for Tablet PCs in the classroom	8
1.4.4 Analysis of instructor whiteboard usage using the Tablet PC	8
1.4.5 Analysis of student note-taking practices using Table PCs	9
1.4.6 Dashboard design	9
1.5 Technology Timeline	10
Chapter 2 Background and Related Work	15
2.1 Introduction	15
2.2 Didactic versus Constructivist Teaching	15
2.3 Computers in the Classroom	16
2.4 Problems with Conventional Technology	19
2.5 eClass (Formerly Classroom 2000)	20
2.6 Tablet-based Instruction and Research	21
2.7 Tablet PC Systems	23
2.8 Overview of Tablet PC Systems for Education	24
2.8.1 UW Classroom Presenter	24
2.8.2 LiveNotes	31
2.8.3 DyKnow	32
2.8.4 MIT InkBoard	34
Chapter 3 The Potential for Tablet PCs	36
3.1 Introduction	36
3.2 A Brief History	37
3.3 Definition and Distinction	38
3.3.1 Active and passive digitizer	38
3.3.2 Slate, convertible, and hybrid	39
3.3.3 Rich ink and digital ink	41
3.3.4 The pen and paper modality	42
3.4 Common Myths	44
3.4.1 Ink-to-text conversion	44
3.4.2 Tablets are not as powerful as a “real” computer	45

3.4.3	Difficult to use	45
3.5	Exploration of Note-taking on the Tablet PC	46
3.5.1	Writing from scratch—starting from a blank “slate”	47
3.5.2	Slide synchronization of instructor white-board contents	48
3.5.3	Association of interactions	52
3.6	Instructor Role	53
3.6.1	Example of whiteboard distribution	53
3.6.2	Points of emphasis	58
3.6.3	Recording the teaching process	59
3.7	Siebel Center Grand Opening	62
3.8	Conclusion	65
Chapter 4	2002 Pilot Study	66
4.1	Introduction	66
4.2	The e-Fuzion System	68
4.2.1	Motivation and need	68
4.2.2	e-Fuzion instructor edition v. 1.0	71
4.2.3	Raster images and vector graphics	73
4.2.4	Fall 2001 and spring 2002 semester experiment	75
4.2.5	Student and instructor in a networked classroom—summer 2002	77
4.3	Determining the Effectiveness of Learning Technology	86
4.4	Research Procedure and Design	87
4.4.1	Subjects	89
4.4.2	Instruments	90
4.4.3	Data analysis	90
4.5	Results	91
4.5.1	Learning differences of students in control and experimental groups	91
4.5.2	Students’ perception and usage of the software	92
4.5.3	The relationship among students’ learning, perception, and interaction	94
4.6	Conclusion	95
4.7	Implications and Limitation	96
Chapter 5	Instructor-only Usage	102
5.1	Introduction	102
5.2	Overview of Instructor Usage	102
5.3	Application Interface Design	103
5.3.1	New design considerations for the Tablet PC	113
5.4	Classroom Survey Evaluation Results	119
5.5	Data Analysis	122
5.5.1	Basic summary information	127
5.5.2	Variations between pages	137
5.5.3	Comparison between classes	143
5.5.4	Changes over time for the semester	153
5.5.5	Level of granularity of analysis	157
5.6	Conclusion	158
Chapter 6	Spring 2007 Pilot Study – Supporting Self-paced Instruction	160
6.1	Introduction	160
6.2	Assessment and Objectives	162
6.2.1	Classroom structures	162
6.2.2	Pre-flight	170
6.2.3	Content presentation	174
6.2.4	Assessments	179
6.2.5	Post-flight	180

6.3	Explicit Polling of Students	184
6.3.1	Teacher display	187
6.4	Student Evaluations	187
6.5	System Implementation	190
6.5.1	System architecture	190
6.5.2	Wireless networking challenges	191
6.5.3	Multicast solutions to networking demands	192
6.5.4	The Pragmatic General Multicast	192
6.5.5	The Sharecast® networking protocol	193
6.5.6	Sharecast® in the spring 2007 pilot study	194
6.6	Conclusion	195
Chapter 7	Fall 2007 and Spring 2008 Dashboard Study	197
7.1	Introduction	197
7.2	Related Work	198
7.3	The Design Space of Visualizations to Support Tablet PC Use	199
7.4	Preliminary Design and Evaluation	200
7.5	Design Approaches	204
7.5.1	User interface	204
7.5.2	Data collection	208
7.6	Dashboard Design	210
7.6.1	Summary: actual work vs. derived data	211
7.6.2	Selection: entire class vs. selected students	212
7.6.3	Time interval: data aggregation period	213
7.6.4	Profile: the use of historical data for comparisons	213
7.7	New Design Approaches	214
7.8	Evaluation	220
7.8.1	CS125 TPC survey results	221
7.8.2	Algebra 1 survey results	222
7.8.3	Algebra 2 survey results	223
7.8.4	Geometry survey results	224
7.9	The Larger Use Design Space	227
7.10	Conclusion	228
Chapter 8	Conclusion	230
Appendix A	Student Surveys	233
A.1	Chapter 5 Survey Questions and Results	233
A.2	Chapter 6 Survey Questions and Results	246
A.3	Chapter 7 Survey Questions and Results	251
Appendix B	Instructor Reflections	298
B.1	Larry DeBrock, Instructor Reflections of Tablet PC Usage	298
B.2	Jason Zych, Instructor Reflections of Whiteboard Usage	299
B.3	Craig Russell, Algebra 2 Instructor	301
B.4	Manish Agrawal, Teaching Assistant for CS125TPC	304
Appendix C	e-Fuzion Documents	305
C.1	Summer 2002 Documentation	305
C.1.1	Contents	305
C.1.2	Installation	306
C.1.3	Linux, Unix, Mac OSX	306
C.1.4	Using e-Fuzion	306
C.1.5	Troubleshooting	311

C.2	e-Fuzion Vector Graphics Library Write Up	312
C.2.1	Introduction	312
C.2.2	Why not use the Dia® or PowerPoint® libraries?	312
C.2.3	It takes all system types	313
C.2.4	VGraphic, simplicity through complexity	313
C.2.5	VGInterface: it doesn't start with 'I'.	314
C.2.6	Conclusion	315
Appendix D	Whiteboard Lecture File Examples	316
D.1	Cinda Heeren – CS225	316
D.2	Algebra 2 Lecture Files	332
D.3	Spring 2007 Lecture Documents	338
Appendix E	Example Log Files	349
E.1	Spring 2005 Instructor Log File	349
E.2	Spring 2007 Pilot Study	350
References	354
Curriculum Vitae	365

List of Tables

4.1	Students' GPA and the number of CS classes taken	89
4.2	Students' demographic information	89
4.3	Sample questions to measure students' perceptions	90
4.4	Learning of students in 2002 and 2001 classes	92
4.5	Learning differences between 2002 and 2001 classes	92
4.6	Students' perception toward the learning software	93
4.7	The relationship among students' learning, perception, and the usage of the system	94
5.1	Spring 2005 teaching schedule	124
5.2	Comparison of spring 2005 total lecture attributes	129
5.3	Comparison of spring 2005 average lecture attributes	129
5.4	Comparison of spring 2005 lecture stroke attributes	130
5.5	Comparison of spring 2005 lecture time attributes in minutes	130
5.6	Comparison of spring 2005 stroke <i>length</i> (in inches) and <i>speed</i> attributes	130
5.7	Unit conversion comparisons of <i>ink</i> and <i>laser</i> length	130
5.8	Comparison of spring 2005 and 2006 CS225 lecture attributes	137
5.9	Ink stroke <i>length</i> , <i>pressure</i> , and <i>speed</i> values for pages characterized as diagrammatic	139

List of Figures

1.1	Diagram outlining the chapters of the dissertation	3
1.2	Tablet PC hardware integration timeline	11
1.3	e-Fuzion applications used from spring 2001 to summer 2003	12
1.4	e-Fuzion application used from fall 2004 to fall 2006	13
1.5	NuPaper® applications used in the spring 2007 to spring 2008 pilot studies	14
2.1	Constructivist and Didactic teaching models	16
2.2	eClass learning environment	20
2.3	An example of an eClass automated archival of a class session	22
2.4	UW Classroom Presenter 3	25
2.5	Ubiquitous Presenter	26
2.6	Ubiquitous Presenter non-Tablet PC interface	27
2.7	Web-based interface for viewing student blogger’s ink annotations	27
2.8	Classroom Learning Partner example application components	29
2.9	Group Learning Project example application components	30
2.10	Elementary Classroom Partner example student submissions	31
2.11	LiveNotes implementation in C# for the Tablet PC.	32
2.12	Screenshot of LiveNotes	33
2.13	LiveNotes application example components	33
2.14	DyKnow Vision	34
2.15	MIT Inkboard interface	35
2.16	Ink board time line feature.	35
3.1	Hovering with the mouse provides different bits of information	39
3.2	Example of student taking an exam on a Tablet PC	40
3.3	Three types of Tablet PCs.	41
3.4	Comparison of ink created with an active and passive digitizer	41
3.5	Comparison of two students writing	42
3.6	CS125 spring 2007 final examination excerpt small	43
3.7	CS125 spring 2007 final examination excerpt large	43
3.8	Fujitsu Slate Tablet PC in a docking station	45
3.9	Examples of in-class use of the Tablet PC	46
3.10	Tablet PC writing sample created by a student observer	47
3.11	Example of student doodling during an examination	49
3.12	Example of lesson objectives	49
3.13	Example of copying instructor-created slide content	50
3.14	Example of student private whiteboard	51
3.15	Siebel Center grand opening classroom of the future lecture demonstration	54
3.16	Example of slide synchronization	55
3.17	Example of synchronized instructor annotations	55
3.18	Original “pre-loaded” whiteboard document	56

3.19	The state of two whiteboards without sending instructor-created annotations	57
3.20	Example of “ink collisions” when sending instructor annotations to student whiteboards . . .	57
3.21	An example of color emphasis of subsequent method calls	58
3.22	Example of duplicating whiteboard contents	59
3.23	Tick tack toe board at the finish state	61
3.24	Example of a time-based algorithm for creating post-lecture slides	63
3.25	Example of ink stroke-based algorithm for tick tack toe	63
3.26	Snapshots from the Siebel center grand opening classroom of the future demonstration	64
4.1	Siebel Center for Computer Science Large lecture classroom settings	67
4.2	Sony VAIO® Slimtop™ Pen Tablet PC model PCV-LX900	68
4.3	The Microsoft® NetMeeting Application	69
4.4	An excerpt from lecture notes created in Microsoft® NetMeeting’s whiteboard application .	70
4.5	Six frames derived from a screen recording of the Microsoft® NetMeeting whiteboard	71
4.6	e-Fuzion instruction edition with textual input area	72
4.7	TeachIT 2000 technology conference e-Fuzion showcase	72
4.8	An illustration of two networking routing schemes.	73
4.9	Comparison of raster image and vector graphic	74
4.10	Handheld computing devices	75
4.11	e-Fuzion client application interfaces for the fall 2001 and spring 2002 semesters	76
4.12	e-Fuzion on-the-fly poll	77
4.13	e-Fuzion instructor control for creating on-the-fly instant polls	77
4.14	Laptops used in the summer 2002 pilot study	78
4.15	e-Fuzion login dialogue	78
4.16	e-Fuzion options	79
4.17	e-Fuzion instructor edition	80
4.18	e-Fuzion client edition	81
4.19	e-Fuzion graphical newsgroup	82
4.20	e-Fuzion interface for sending text-based question and requiring a graphical response	83
4.21	e-Fuzion interface for sending graphical illustrations and multiple-choice question response. .	83
4.22	e-Fuzion poll results display	84
4.23	Illustration of textual, graphical, and multiple-choice poll	84
4.24	e-Fuzion poll builder	85
4.25	e-Fuzion display from the perspective of a student	86
4.26	Snapshots of student screens	98
4.27	student moderately off-task	99
4.28	Student completely off task	100
5.1	Pre-logging version of e-Fuzion	104
5.2	e-Fuzion instructor edition	105
5.3	e-Fuzion with logging and reduced user interface	106
5.4	Revision of e-Fuzion 2005	108
5.5	e-Fuzion 2005 Version 2 stand alone application	109
5.6	e-Fuzion login screen	110
5.7	e-Fuzion 2005 version 3	111
5.8	e-Fuzion Stand alone client	112
5.9	e-Fuzion 2006 Instructor-only use	113
5.10	NuPaper Classroom® standalone presentation application.	114
5.11	Excerpt of NetMeeting® Whiteboard document	115
5.12	Digital ink magnification	115
5.13	Comparison of digital ink technologies	115
5.14	Instructor whiteboard examples	116
5.15	Example of whiteboard usage	117
5.16	An excerpt of a data structures lecture	118

5.17	An excerpt of an introductory lecture in Computer Science	118
5.18	Survey question one	120
5.19	Survey question nine	121
5.20	Survey question 24	121
5.21	Survey question five	122
5.22	Example log file used to capture application usage in the spring 2005 semester	124
5.23	Lecture summarization of whiteboard usage	125
5.24	126
5.25	126
5.26	Lecture summary information	128
5.27	CS225 lecture six whiteboard document	128
5.28	An example of too many “inked” annotations	131
5.29	Average number of laser tool strokes	132
5.30	Average number of ink strokes created	133
5.31	Average stroke deletion count	134
5.32	Inking time per lecture	135
5.33	Average erasure time	135
5.34	Average laser tool time	136
5.35	Average ink stroke speed	137
5.36	Average ink stroke length	138
5.37	Measurement of ink density	139
5.38	Pages assigned a diagrammatic ranking of one	140
5.39	Pages assigned a diagrammatic ranking of two	140
5.40	Pages assigned a diagrammatic ranking of three	141
5.41	Pages assigned a diagrammatic ranking of four	141
5.42	Pages assigned a diagrammatic ranking of five	142
5.43	Comparison of instructor ink stroke speed	143
5.44	Page navigations in the spring 2005 CS225 12PM lectures	144
5.45	Page visitations for two lectures	145
5.46	A Comparison of page visitations in two CS225 lectures	146
5.47	Summary information for CS125 lectures for February 11, 2005	147
5.48	Pages one to four of the CS125 whiteboard documents for February 11, 2005	147
5.49	Pages five to nine of the CS125 whiteboard documents for February 11, 2005	148
5.50	Total time allocation for the 2PM CS125 lectures on February 11, 2005	149
5.51	Summary information for CS125 lectures for March 14, 2005	150
5.52	Pages one to six of the CS125 lecture whiteboard documents for March 14, 2005	151
5.53	Pages seven to thirteen of the CS125 lecture whiteboard documents for March 14, 2005	151
5.54	Page visitation for CS125 lectures on March 14, 2005	151
5.55	Total time allocation for CS125 lectures on March 14, 2005	152
5.56	Summary information for CS225 lectures for March 14, 2005	152
5.57	CS225 whiteboard documents for March 14, 2005	153
5.58	Speed variation for CS225 lectures on March 14, 2005	154
5.59	Average laser tool usage per lecture	154
5.60	Measurement of ink stroke pressure	158
5.61	Summary of ink stroke speed	159
5.62	Length of all ink strokes for a single page	159
6.1	The SLICE educational technologies laboratory room setup	161
6.2	Student note excerpts from CS125 fall 2006	162
6.3	This matrix is an attempt to spatially model representations of classroom structures	164
6.4	The contents of an example PDF document for one objective of a lesson	165
6.5	Flow chart illustrating the different states of a classroom structure	169
6.6	Dialogue box displaying the status of lecture objectives	170

6.7	First design of our spring 2007 student user interface	171
6.8	Example teacher’s display	172
6.9	Physics 101 course syllabus at the University of Illinois at Urbana-Champaign	173
6.10	Initial pre-flight user interface design	175
6.11	Objective monitor control	175
6.12	Example pre-flight question	176
6.13	User interface for note-taking mode in the spring 2007 pilot study	177
6.14	Modified interface for the summer 2007 CS225 pilot study	178
6.15	Summer 2007 application interface	179
6.16	Example of an assessment for the C++ programming objective	181
6.17	Updated design (on the right) of post-flight assessment user interface (on the left)	182
6.18	Post-flight interface used in the spring 2007 pilot study	183
6.19	Explicit feedback poll sent to students in note-taking mode	185
6.20	Template for explicit feedback poll results	185
6.21	Explicit feedback poll and results	186
6.22	Example of “drilling down” into explicit poll results	187
6.23	An example of our first teacher’s dashboard design	188
6.24	Spring 2007 pilot study survey results for questions 2–5, 11–18, and 23–24	189
6.25	Illustration of the system architecture for the spring 2007 pilot study	191
6.26	Multicast transmission illustration	192
6.27	Illustration of automatic retransmission of packets sent over a wireless network	193
6.28	Illustration of forward error correction (FEC) mechanism for packet delivery	193
6.29	Illustration of bandwidth throttling	194
6.30	The XML schema contains three objectives and expects three PDF files	195
6.31	Sharecast® network console	195
7.1	Motivating the Teacher’s Dashboard	201
7.2	Fall 2007 VNC-based display	203
7.3	An arrangement of 16 VNC student windows	204
7.4	VNC implementation by NetSupport™	205
7.5	NuPaper Study document dialogue	208
7.6	Illustration of the NuPaper Classroom® examination interface	209
7.7	Example of an ink stroke’s attributes recorded in the log files	210
7.8	Thumbnails in the file explorer of Windows Vista	212
7.9	Dashboard display spanning two monitors	214
7.10	Front view of instructor using a two-monitor dashboard display	215
7.11	Dashboard display spanning four monitors	216
7.12	NuPaper® Dashboard display	217
7.13	NuPaper® Dashboard display	218
7.14	NuPaper® student selected whiteboard	219
A.1	Chapter 5 survey question 1	235
A.2	Chapter 5 survey question 2	235
A.3	Chapter 5 survey question 3	235
A.4	Chapter 5 survey question 4	236
A.5	Chapter 5 survey question 5	236
A.6	Chapter 5 survey question 6	237
A.7	Chapter 5 survey question 7	238
A.8	Chapter 5 survey question 8	238
A.9	Chapter 5 survey question 9	239
A.10	Chapter 5 survey question 10	239
A.11	Chapter 5 survey question 11	240
A.12	Chapter 5 survey question 12	240
A.13	Chapter 5 survey question 13	240

A.14 Chapter 5 survey question 14	241
A.15 Chapter 5 survey question 15	241
A.16 Chapter 5 survey question 16	242
A.17 Chapter 5 survey question 17	242
A.18 Chapter 5 survey question 18	242
A.19 Chapter 5 survey question 19	243
A.20 Chapter 5 survey question 20	243
A.21 Chapter 5 survey question 21	244
A.22 Chapter 5 survey question 22	244
A.23 Chapter 5 survey question 23	245
A.24 Chapter 5 survey question 24	245
A.25 Chapter 8 survey question 1 – CS125TPC	252
A.26 Chapter 8 survey question 2 – CS125TPC	253
A.27 Chapter 8 survey question 3 – CS125TPC	253
A.28 Chapter 8 survey question 4 – CS125TPC	254
A.29 Chapter 8 survey question 5 – CS125TPC	254
A.30 Chapter 8 survey question 6 – CS125TPC	255
A.31 Chapter 8 survey question 7 – CS125TPC	255
A.32 Chapter 8 survey question 8 – CS125TPC	256
A.33 Chapter 8 survey question 9 – CS125TPC	256
A.34 Chapter 8 survey question 10 – CS125TPC	257
A.35 Chapter 8 survey question 11 – CS125TPC	257
A.36 Chapter 8 survey question 12 – CS125TPC	258
A.37 Chapter 8 survey question 13 – CS125TPC	259
A.38 Chapter 8 survey question 14 – CS125TPC	260
A.39 Chapter 8 survey question 15 – CS125TPC	260
A.40 Chapter 8 survey question 16 – CS125TPC	261
A.41 Chapter 8 survey question 17 – CS125TPC	261
A.42 Chapter 8 survey question 1 – algebra 1	262
A.43 Chapter 8 survey question 2 – algebra 1	262
A.44 Chapter 8 survey question 3 – algebra 1	263
A.45 Chapter 8 survey question 4 – algebra 1	263
A.46 Chapter 8 survey question 5 – algebra 1	264
A.47 Chapter 8 survey question 6 – algebra 1	264
A.48 Chapter 8 survey question 7 – algebra 1	265
A.49 Chapter 8 survey question 8 – algebra 1	265
A.50 Chapter 8 survey question 9 – algebra 1	266
A.51 Chapter 8 survey question 10 – algebra 1	266
A.52 Chapter 8 survey question 11 – algebra 1	267
A.53 Chapter 8 survey question 12 – algebra 1	267
A.54 Chapter 8 survey question 13 – algebra 1	268
A.55 Chapter 8 survey question 14 – algebra 1	269
A.56 Chapter 8 survey question 15 – algebra 1	269
A.57 Chapter 8 survey question 16 – algebra 1	270
A.58 Chapter 8 survey question 17 – algebra 1	270
A.59 Chapter 8 survey question 1 – algebra 2	271
A.60 Chapter 8 survey question 2 – algebra 2	271
A.61 Chapter 8 survey question 3 – algebra 2	272
A.62 Chapter 8 survey question 4 – algebra 2	272
A.63 Chapter 8 survey question 5 – algebra 2	273
A.64 Chapter 8 survey question 6 – algebra 2	273
A.65 Chapter 8 survey question 7 – algebra 2	274
A.66 Chapter 8 survey question 8 – algebra 2	274

A.67 Chapter 8 survey question 9 – algebra 2	275
A.68 Chapter 8 survey question 10 – algebra 2	275
A.69 Chapter 8 survey question 11 – algebra 2	276
A.70 Chapter 8 survey question 12 – algebra 2	276
A.71 Chapter 8 survey question 13 – algebra 2	277
A.72 Chapter 8 survey question 14 – algebra 2	278
A.73 Chapter 8 survey question 15 – algebra 2	278
A.74 Chapter 8 survey question 16 – algebra 2	279
A.75 Chapter 8 survey question 17 – algebra 2	279
A.76 Chapter 8 survey question 1 – algebra 1 without technology	280
A.77 Chapter 8 survey question 2 – algebra 1 without technology	280
A.78 Chapter 8 survey question 3 – algebra 1 without technology	280
A.79 Chapter 8 survey question 4 – algebra 1 without technology	281
A.80 Chapter 8 survey question 5 – algebra 1 without technology	282
A.81 Chapter 8 survey question 6 – algebra 1 without technology	282
A.82 Chapter 8 survey question 7 – algebra 1 without technology	282
A.83 Chapter 8 survey question 8 – algebra 1 without technology	283
A.84 Chapter 8 survey question 9 – algebra 1 without technology	283
A.85 Chapter 8 survey question 10 – algebra 1 without technology	284
A.86 Chapter 8 survey question 11 – algebra 1 without technology	284
A.87 Chapter 8 survey question 12 – algebra 1 without technology	285
A.88 Chapter 8 survey question 1 – geometry without technology	285
A.89 Chapter 8 survey question 2 – geometry without technology	285
A.90 Chapter 8 survey question 3 – geometry without technology	286
A.91 Chapter 8 survey question 4 – geometry without technology	286
A.92 Chapter 8 survey question 5 – geometry without technology	287
A.93 Chapter 8 survey question 6 – geometry without technology	287
A.94 Chapter 8 survey question 7 – geometry without technology	287
A.95 Chapter 8 survey question 8 – geometry without technology	288
A.96 Chapter 8 survey question 9 – geometry without technology	288
A.97 Chapter 8 survey question 10 – geometry without technology	289
A.98 Chapter 8 survey question 11 – geometry without technology	289
A.99 Chapter 8 survey question 12 – geometry without technology	290
A.100 Chapter 8 survey question 1 – geometry	290
A.101 Chapter 8 survey question 2 – geometry	291
A.102 Chapter 8 survey question 3 – geometry	291
A.103 Chapter 8 survey question 4 – geometry	292
A.104 Chapter 8 survey question 5 – geometry	293
A.105 Chapter 8 survey question 6 – geometry	293
A.106 Chapter 8 survey question 7 – geometry	294
A.107 Chapter 8 survey question 8 – geometry	294
A.108 Chapter 8 survey question 9 – geometry	295
A.109 Chapter 8 survey question 10 – geometry	296
A.110 Chapter 8 survey question 11 – geometry	296
A.111 Chapter 8 survey question 12 – geometry	297

Chapter 1

Introduction and Motivation

1.1 Introduction

This thesis is about how to use Tablet PCs to enhance instruction and facilitate educational research. Tablet PCs have the following unique combination of traits:

They *can be used* in a classroom. Because keyboards are noisy, traditional laptops cannot be used by all students in a classroom without being a distraction. And because much of what professors draw on the whiteboard, and students write in their notes, is graphical and pictorial, typing is an insufficiently rich input method. Tablets eliminate these problems and therefore, to repeat, *can be used* in classrooms.

What students and instructors write can be recorded. This allows students to see the professor's notes after class and to easily share their notes. Perhaps more importantly, it can allow for post-hoc analysis, whether for teachers to review their own and their students' writing, or for research purposes to obtain a microscopic view of the class—something never before possible.

What students write can be seen by the instructor in real-time. In small classes, teachers will often walk about the room to see what students are writing. If the students are using Tablet PCs, this writing can be sent directly to the teacher's computer, allowing the teacher to monitor more students. In a large class, this kind of review may become possible where it never has been before.

Teacher and students can communicate more effectively. This ability can be used in numerous ways, including having students answer poll questions (improving on the "showing of hands" method), allowing students to ask questions without interrupting the class, providing a summary of the level of student engagement, and countless others.

This thesis presents a set of studies that touch on these points in varying ways. In chapter 2 we present related works. Chapter 3 introduces the Tablet PC device, highlighting the technical innovation and educational possibilities. In chapter 4 we discuss a study we did in summer 2002, just before the arrival of

the Tablet PC device. Chapter 5 presents a summary of instructor-only usage of Tablet PCs, using our e-Fuzion software, as a glorified whiteboard hooked up to a projector display, and presents data on that usage. Chapter 6 shows how using Tablet PCs in the classroom can allow for fundamentally new classroom structures, describing the spring 2007 experimental section of our introductory programming course (CS125). Chapter 7 introduces the capstone Teacher’s Dashboard study performed in the Fall 2007 and Spring 2008 semesters. In this chapter we present a formalized design process for building different dashboard prototypes for in-class use.

Figure 1.1 presents a pictorial representation of the organization of this dissertation. Chapter 1 motivates the need for educational technology that can be integrated into a classroom learning environment. Chapter 2 sets the scene by providing a quick overview of the relevant literature and chapter 3 introduces the Tablet PC. After experimenting with non-custom software, we began developing e-Fuzion, first for laptops (presented in chapter 4) and then pen-input devices, culminating in the redesign and development of a Tablet PC version of e-Fuzion (chapter 5). Beginning with chapter 5 all of the work presented in this dissertation, as described in the previous paragraph, incorporated Tablet PC usage.

1.2 Motivation

Various studies [1, 13, 15, 29, 46, 48, 58, 66, 72, 79, 81, 93, 97, 113, 119, 128, 130, 137, 139, 140, 150, 166, 168, 172, 184, 190] have made a strong case that wireless technology that allows real-time sharing and exchange of information by instructors and/or students offers benefits to students’ performance in, and enjoyment of, classes in a range of subjects. However, many other studies suggest that technology is falling short of the ambitious goals that technology enthusiasts and pioneers in this field felt were within easy reach (e.g. Cuban’s seminal work *Oversold and Underused: Computers in the Classroom* [44].) In an almost comedic retort to the technophiles, Clifford Stoll’s *High Tech Heretic: Why Computers Don’t Belong in the Classroom and Other Reflections by a Computer Contrarian* offers keen insight [177]. One reason for the disappointments, observed by most pedagogues, is that many of these studies focus on the engineering problem of developing and delivering individual tools, rather than on the underlying—and much harder to define—problem of integrating technology into the learning environment (the teaching curriculum) in a way that reinforces and builds on pedagogical goals [42, 59, 112, 154, 156, 163]. On the other hand Wenglinsky presents quantitative information based on an analysis of data from the 1996 and 2000 National Assessment of Educational Progress (NAEP) (test scores from more than 40,000 students) to show a positive association between science and mathematics scores and the integration of technology in constructivist-based instruction

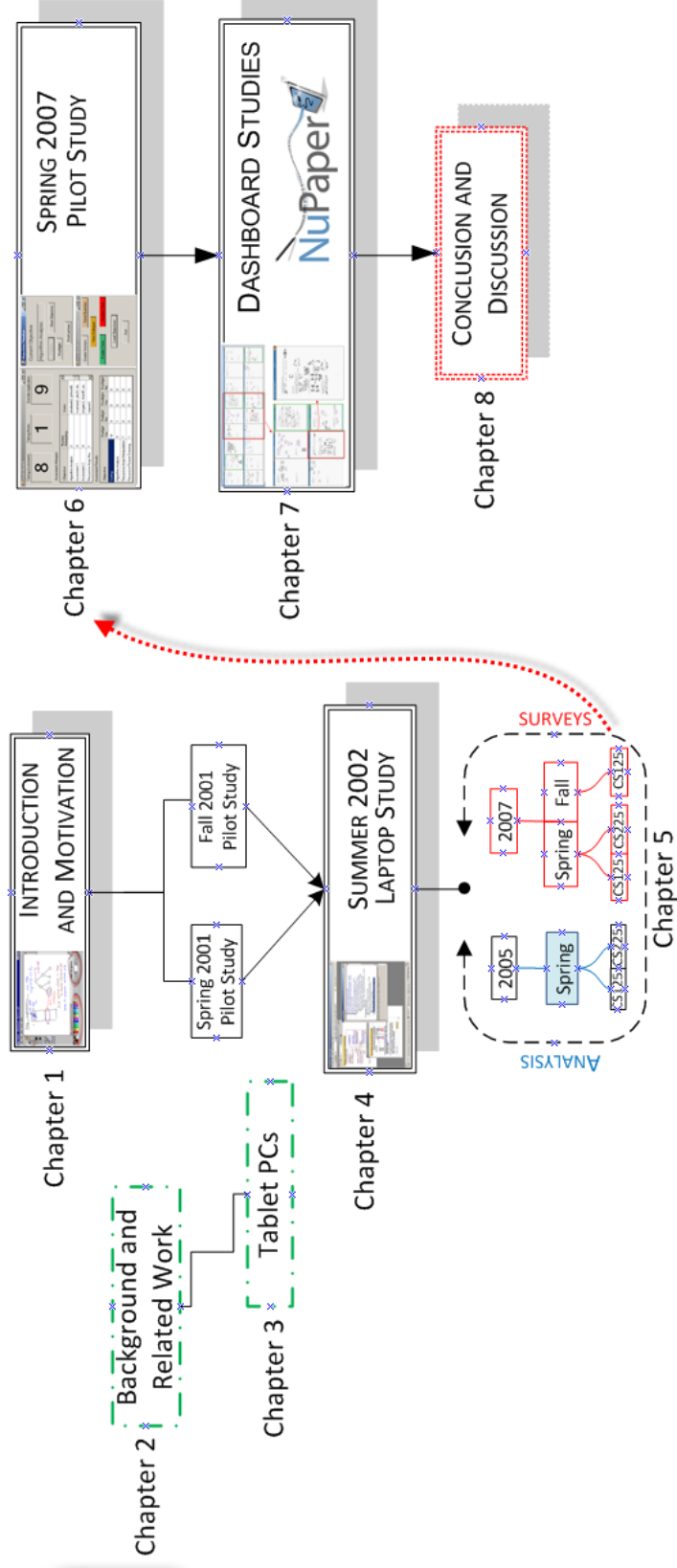


Figure 1.1: Diagram outlining the chapters of the dissertation

[194].

The study of Tablet PC technology in learning environments, coupled with the development and implementation of classroom monitoring systems, has the potential to change the way we teach. Educators at all levels will be able to increase interaction and, more importantly, will be able to monitor learner behaviors to identify students having difficulty or becoming bored. This feedback, if reliable and readily available, could provide instructors with an insight into their students' level of engagement. In addition, students' in-class experience will be greatly improved by allowing opportunities for interaction with course content. For example, note-taking, performing assessments, interacting with peers, and providing a way to more easily articulate complex concepts may all be enhanced by this technology. The power of Tablet PC technology can bring about new teaching practices and new levels of engagement for students and teachers.

A challenging task for most, if not all, instructors, is gauging student interest and learning during the course of a lecture. Under current approaches, it is very difficult to understand how well the students are grasping the lecture while the lecture is going on. It is unrealistic to require one individual to observe, assess, and evaluate (monitor) the classroom, while following a strict lesson plan and lecture objectives. The ideal would be to adjust the lecture midstream to help the students. Currently, teachers monitor and assess their students' progress by a variety of mechanisms, such as observation of student participation, listening to student questions and comments, providing in-class exercises, and of course on a larger scale, graded assessments.

With the help of technology, instructors will be able to better assess and monitor their students inside and outside of the classroom. Studies have shown convincingly that a teacher's use of formative assessments is a crucial determinant of students' learning outcomes [19, 125]. Currently, the main method of classroom assessment is limited interaction with students. Another type of assessment is passive, or unobtrusive. This is the type of assessment that a teacher does by simply looking around the class, or by walking around the room looking at students' individual work and body language such as facial expressions. The best teachers excel at this, but it is an uncertain undertaking, difficult to quantify, hard to learn, and applicable mainly in classrooms where the teacher knows the students very well. We believe the Tablet PC can dramatically enhance the college teacher's ability to unobtrusively [193] assess a class.

In this work, we have developed and explored a range of prototype tools designed to meet specific needs identified by instructors to improve instructor delivery and student retention of class material. These tools were extremely successful, based on qualitative feedback from study participants. However, as we experienced in our 2002 summer pilot study (presented in chapter 4 section 4.2.5) even though student engagement improved over the baseline (i.e. without the technology) the new technologies failed to engage

a significant percentage of students. Instructor participants and administrative personnel also observed this phenomenon. This observation is not unprecedented; it echoes that made by numerous studies [41, 42, 93, 151, 160, 181, 205].

We were motivated to study the possibilities for Tablet PC use in the classroom. Specifically, we have addressed these issues:

- How do instructors use the Tablet PC while lecturing?
- How can we use the Tablet PC to study student note-taking?
- Does the Tablet PC allow for a class to be structured in a fundamentally new way?
- How can the Tablet PC be used to enhance the transparency of the class, and how will teachers use the information it provides?

1.3 Overview of Dissertation

In this dissertation, we present a number of studies piloted over a seven year time span beginning in the fall semester 2001 and concluding in spring 2008. Throughout our studies we have investigated and experimented with ways to enhance the learning experience of our students. Almost all research examining the role of the classroom teacher in the process of educating students has come to the same conclusion: the key to increased learning is effective instruction [123]. Effective instruction involves, among other things, having tools which are appropriate to the demands of the particular learning environment.

Chapter 2: We summarize the body of literature related to the work presented in this dissertation.

Chapter 3: An introduction to the Tablet PC and a discussion of its potential are presented.

Chapter 4: A study done prior to the advent of Tablet PCs—using laptops and an earlier incarnation of pen-enabled computer—is presented. The use of computers to aid in-class communication is explored.

Chapter 5: The use of Tablet PCs to record instructor’s actions for later analysis is discussed. Taking classes from several semesters, we present some data about instructor writing that could not, in any practical way, be obtained without them. Since the instructors use the Tablet PC just as if it were an overhead projector, the results would seem to be transferable to the vast number of instructors who use such devices. (The transferability of these results to blackboard instruction is more questionable, because writing on a blackboard is mechanically much different from writing on an overhead projector or Tablet PC.) Thus, the Tablet PC provides a kind of laboratory for the study of instructor writing. We ask simple questions such as:

1. How much time does an instructor spend writing in a lecture, on a topic, page, or, over the course of an entire semester? How do these measurements compare among multiple sections an instructor teaches of the same class?
2. How fast does an instructor write on average or before navigating to a new slide, topic, or document?
3. How much time does an instructor spend using the laser pointer and which pages of a document have the most laser pointer usage?

Chapter 6: By enhancing communication, Tablet PCs can increase the instructor’s ability to monitor and control the class. The perhaps counter-intuitive result is that this allows the instructor to create a *highly flexible* classroom structure. With the limited communication capabilities of a regular classroom, an instructor can allow only a very limited amount of self-pacing before the classroom becomes chaotic. With networked Tablet PCs—plus a good deal of before-class preparation—the instructor can *know where every student is*. With this vital information, she can tailor discussion to those in the majority (or to those who are lagging, or who are leading, or whatever group she chooses); she can tell what material is giving the most difficulty; she can encourage the leading students to help the lagging students (hopefully to the benefit of both). In this chapter, a class given in the spring 2007 semester, which made use of this capability, is described.

Chapter 7: An instructor always wants to know, above all, when the students are learning and what they are struggling with. In a conventional classroom, an instructor will ask questions, set in-class exercises, and encourage questions from the students, and use the results to gauge the students’ understanding (and even their level of engagement). The conventional classroom can be enhanced—at no cost to the instructor—by providing a view of each student’s work. In the design of this “teacher’s dashboard,” the critical question is how to select, organize, and display data so that the teacher can quickly peruse it and home in on the interesting parts. In four different high school classes in Fall, 2007, and spring, 2008, a dashboard was deployed and its design, and the effectiveness of that design, were studied.

Throughout this dissertation we refer to a system called e-Fuzion. Motivated by early pen-based input devices, the first version of e-Fuzion was developed in the fall of 2001 by two of our undergraduate students to ease the burden of student note-taking. The major feature that distinguished our system from others at the time was the integration of a vector graphics library. We wanted to build a system that could support seamless integration into our large lecture halls in addition to supporting our off-campus students in our Illinois Internet Computer Science Program (I²CS) where bandwidth for students using dial-up connections

was limited.

1.4 Contributions

This dissertation makes several scientific contributions: The investigation of enhancing education through Tablet PCs will significantly advance the literature base and level of understanding regarding (i) computer-mediated classroom interactions in a wireless classroom, (ii) educational promise of Tablet PC based instructional delivery, note-taking, monitoring, and an illustration of the pedagogical promise of digital ink. The dissertation reports on pilot studies including participants (instructor and student) from secondary and university level classroom settings. These studies address several important research questions.

1.4.1 Assessing Tablet PCs impact on student learning

Throughout our work we have been able to track student learning behaviors and performance levels in support of identifying how students use Tablet PCs to: (i) record and review notes and (ii) complete in-class and homework assignments. We collected interactions on the tablet including exams, assessments, and note-taking through logs recorded by our software. In addition each pilot study includes surveying of students and teachers and live classroom observations. To date, we have approximately three years of data for analysis.

The current literature base contains virtually no empirical research on the impact of Tablet PCs on learning. This is due largely to the recent emergence of Tablet PCs as learning tools and the comparatively small market share that Tablet PCs have relative to their larger desktop counterparts. However, as tablet use increases there will be a need to understand how Tablet PCs can be used most effectively in the learning process. Ultimately, we hope this work will provide a foundation for pedagogical guidelines that can be easily implemented in a classroom of Tablet PCs.

1.4.2 Analysis of student attitudes using Tablet PCs

Our studies generated findings documenting students' note-taking practices and articulation of self expression, creativity, and collaboration as well as gathering their opinions and attitudes regarding the use of Tablet PCs in the classroom. By examining these aspects of learning, in addition to the more performance based measures, researchers and educators will better understand: (i) how students use Tablet PCs in self-paced learning environments, (ii) student attitudes towards Tablet PCs as learning tools, (iii) student perceptions of how Tablet PCs might best be used to help them learn.

In addition to monitoring and diagnosing student learning and performance levels as described in [57, 122,

130, 190], other aspects of student learning were analyzed in relation to Tablet PC usage in the classroom. Given the pen-based capabilities of Tablet PCs, they provide new avenues of self expression [52], creativity [16, 35], and collaboration to students that are not easily available through traditional paper based media. Educators need to understand how students perceive Tablet PCs in general. Do they view them as useful learning tools? Do they perceive tablets affecting their studies in particular ways? Understanding students' attitudes towards Tablet PCs and pen-based learning tools will directly impact the guidelines under which Tablet PCs should be employed.

1.4.3 Analysis of teacher behaviors in accommodating for Tablet PCs in the classroom

This question analyzed the teachers' behaviors and tendencies regarding the integration of tablet computers into their lessons. We have hundreds of surveys performed over the course of our studies where a Tablet PC was used to teach. The earlier surveys led us to believe that we may be able to ascertain (i) the level of tablet computer integration into daily lessons, (ii) the pedagogical strategies, including in-class and homework activities and assessment techniques, used by the teachers, (iii) the level of effectiveness of these various strategies and (iv) student perception of the integration. As with any new medium, there is a learning curve that will be experienced by users; the same applies with tablet computers. By understanding the tendencies, techniques, and levels of success experienced by these innovative educators, other teachers can learn from their experiences and minimize the learning curve they will experience when they first deploy tablets in their classrooms. The data gathered from student and instructor use over the past four years was analyzed and summarized and provides information related to usability.

1.4.4 Analysis of instructor whiteboard usage using the Tablet PC

This question addressed how instructors interact with a Tablet PC. Although there are a number of papers that summarize an instructor's experience using a Tablet PC in the classroom (see section 2.7), no research to our knowledge presents usage information at the granularity of the ink stroke to help characterize an instructor's teaching style within a class session, a semester, or over an extended period of time such as an academic year. Richard Anderson and his team analyzed [8] instructor-drawn diagrams to attempt to differentiate between annotations, different stages of a single diagram (phases), and to construct the active context in a diagram. The idea was that such identification might help to automate the analysis of real-time and post lecture files created on Tablet PC devices as a means of enhancing the learning environment.

Having captured Tablet PC usage from our whiteboard application, we had the opportunity to explore an

instructor’s teaching style through extensive log file analyses. We discovered patterns of usage that support a particular instructor’s teaching style. We present a number of examples in chapter 5 to illustrate how the Tablet PC can provide a summary and an objective view of an instructor’s lecture that was not possible before. We also present a number of survey results collected over the course of three semesters to summarize students’ perception of an instructor’s use of the Tablet PC in class.

1.4.5 Analysis of student note-taking practices using Table PCs

A great deal of research has been performed on note-taking practices [14, 20, 27, 70, 73, 84, 100–105, 198]. See Yin [121] for a good overview of research in note-taking. Prior to the arrival of the Tablet PC it was nearly impossible to monitor traditional student note-taking practices (i.e. students taking notes with paper note books and a pen or pencil) in real-time. Technologies such as video recording devices for capture and video cards to digitize the contents of a student’s notebook were potentially helpful, but were limited. Aside from the expense (each student would require a similar setup), and the obtrusive nature of the collection process (imagine each student with a camera hanging over their shoulder), it remains unclear to what degree the technology could have recorded user interactions with their notebooks. As a result, a number of early tablet-based systems (prior to the arrival of the Tablet PC) were developed to emulate and record note-taking practices [118, 120, 142, 175] in corporate settings.

Our analysis of student note-taking practices on the Tablet PC includes a number of different models:

1. students taking notes from scratch (on a blank whiteboard) while instructed by a teacher **not** using a Tablet PC
2. students taking notes from scratch (on a blank whiteboard) while instructed by a teacher using a Tablet PC
3. students taking notes by annotating instructor-provided presentations on their Tablet PC device (note that this does not necessarily require a networked infrastructure as the instructor presentations could be manually imported to each student’s machine)

1.4.6 Dashboard design

A challenging task for most, if not all instructors, is the task of trying to gauge student interest and learning during the course of a lecture. Under current approaches, it is very difficult to understand how well the students are grasping the lecture while the lecture is in progress. This research question attempts to address how the Tablet PC technology can assist an instructor in the observation, assessing, and evaluation (monitoring) of the classroom. Ideally, the dashboard would enable an instructor to adjust the lecture midstream to help the students. Currently, teachers monitor and assess their students’ progress by a variety

of mechanisms, such as observation of student participation, listening to student questions and comments, providing in-class exercises, and of course on a larger scale, graded assessments. Chapter 7 of this dissertation introduces our design of a system to display the *type*, *level*, and *pace* of student activity in the classroom. For example, the pace attribute might provide the instructor with information whether a student is lagging behind, moving ahead, or following closely with the lesson. How exactly the type, level, and pace are defined will vary among differing classroom structures. As an example, the instructor might want a real-time comparison of the pace of individual students or an aggregate amongst a subset of students for a given activity. A cursory glance of a good dashboard might point an instructor toward the students requiring special attention. Student evaluations of five high school classes and one university class taught in our Educational Laboratory Classroom provided qualitative measurements of instructor utilization of the teacher’s dashboard.

1.5 Technology Timeline

In the course of this research, we have used a variety of technologies. Our hardware began with earlier pen-enabled computers and moved to the Tablet PC. Our software, which is mostly homegrown, has changed every semester. This can lead to some confusion. Therefore, we present a number of graphical illustrations that summarize the evolution of our e-Fuzion application and integration of tablet-based technologies. These illustrations are not mockups, but rather annotated screenshots of the actual application windows.

Figure 1.2 illustrates the timeline integration of hardware used in the pilot studies over the course of our seven year study. The devices listed under “fall 2004 to present” are Tablet PC devices which we introduce in chapter 3. The devices in the figure with a border symbolize that the device contained an “active digitizer” which, as we discuss in section 3.3.1, is the same technology used in the Tablet PC. Dates colored in gray (fall 2001 and fall 2002) represent devices used primarily by instructors while all other devices were used by students.

In the fall 2001 semester the Department of Computer Science acquired its first active digitizer (see section 3.3.1) device when it purchased the Sony VAIO® Slimtop™ Pen Tablet PC, model, PCV-LX900 (figure 4.2). The release of this device was about two years before the arrival of the “Tablet PC” as described in chapter 3. While the pen input provided a more “natural” input modality (when compared to the mouse) for writing on a computer, we lacked software that could take advantage of this new technology. As a result, we began in earnest the development of e-Fuzion, an application initially developed as a presentation tool for instructors and later redesigned as a Tablet PC educational system.

Figure 1.3 contains snapshots of the different e-Fuzion versions used by students and instructor (red

Tablet Integration Timeline



Figure 1.2: Tablet PC hardware integration timeline beginning in the fall 2001 to spring 2008

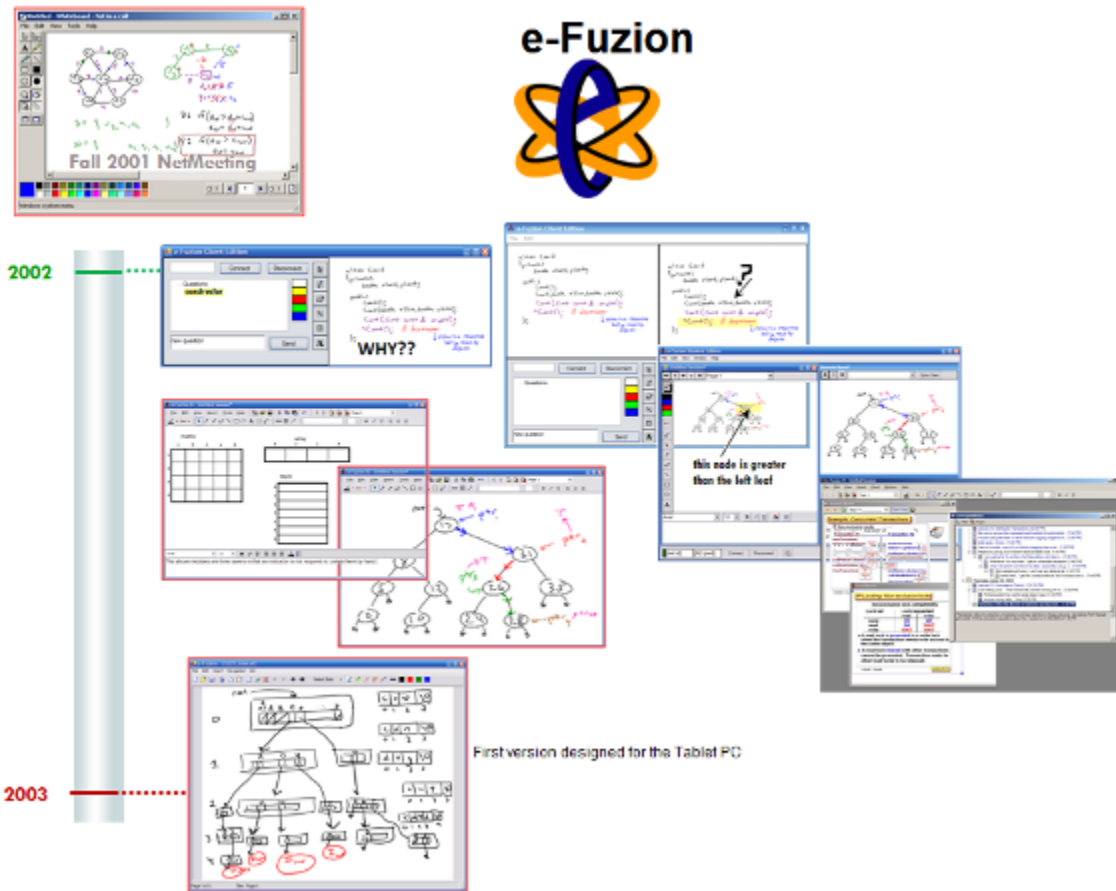


Figure 1.3: e-Fuzion applications used from spring 2001 to summer 2003

border) beginning in the fall 2001 semester (before the first e-Fuzion release) and spanning to the fall 2003 semester.

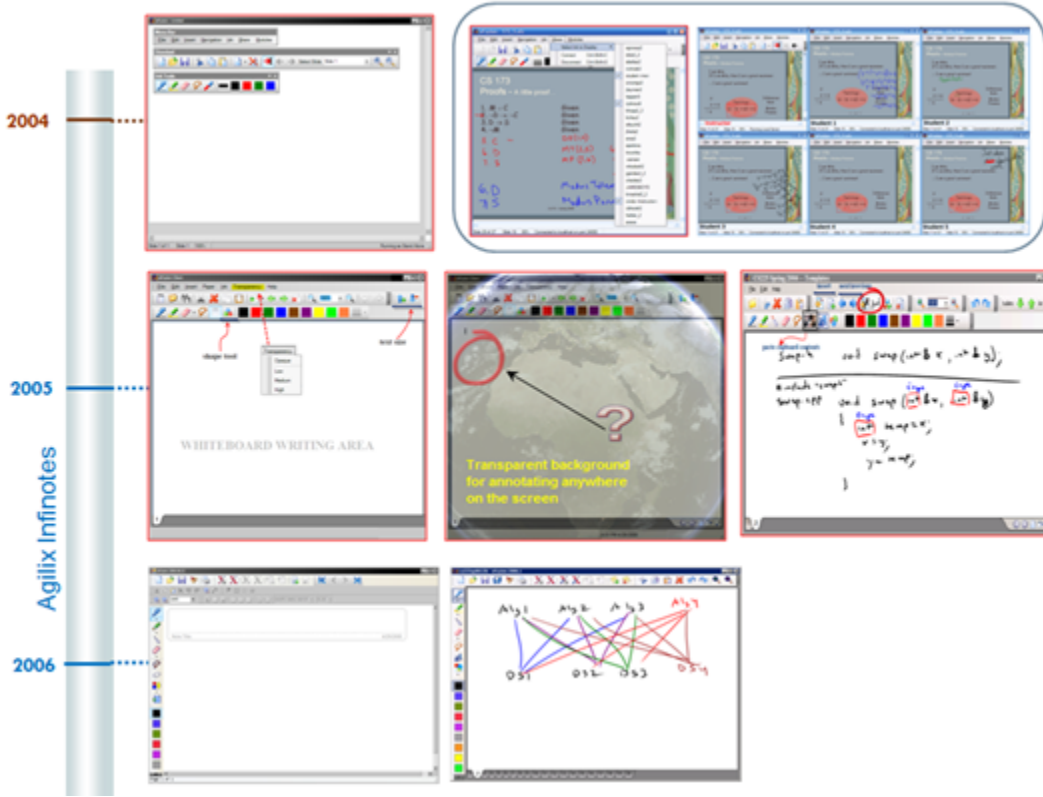


Figure 1.4: e-Fuzion application used from fall 2004 to fall 2006

Figure 1.4 displays all the Tablet PC versions of e-Fuzion which we introduce in chapter 5. Beginning in the spring 2007 semester and ending in the spring 2008 semester we incorporated the use of NuPaper Classroom® written by Boris Capitanu of NuPaper® (figure 1.5), to perform the spring 2007 pilot study (chapter 6) and the fall 2007 and spring 2008 dashboard studies (chapter 7).

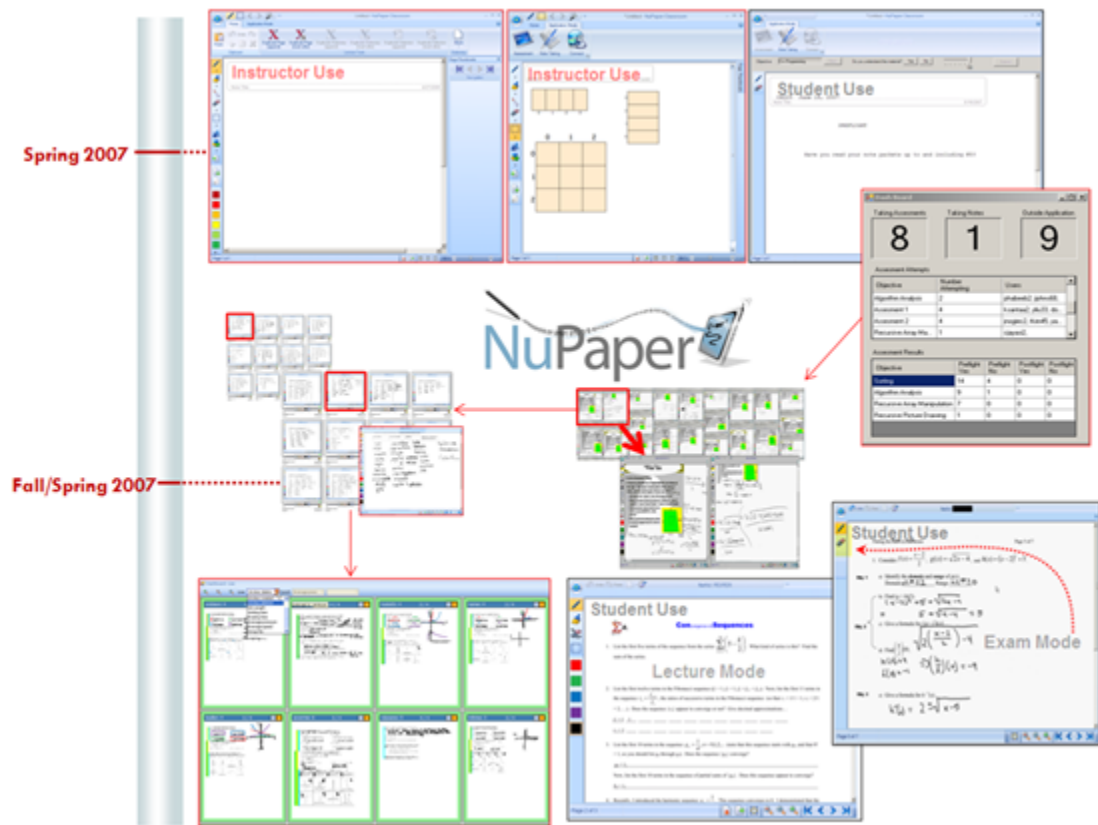


Figure 1.5: NuPaper® applications used in the spring 2007 to spring 2008 pilot studies

Chapter 2

Background and Related Work

2.1 Introduction

In this chapter we introduce the body of literature related to the work presented in this dissertation. First we present general pedagogical practices integrated into the paper world before the personal computer (PC) arrived and attempts were made to integrate it into the classroom. In section 2.3 we present a summary of numerous studies investigating how computers have been integrated into primary, secondary, and college classrooms in the United States. In this overview, we rely heavily on Dr. Larry Cuban's research[41–44]. In section 2.4 we motivate the need for integrating ubiquitous technologies into the learning process from a historical perspective at the University of Illinois at Urbana-Champaign. Section 2.5 summarizes the pioneering work of Abowd and Brotherton [1, 86, 88] completed at Georgia Tech in a project originally called Classroom 2000 and known today as eClass. In section 2.6 we present a synopsis of tablet-based instruction which details the myriad studies performed on tablet-based computers. Although there is a clear distinction between pen-input devices of the past and the Tablet PC (see chapter 5), which integrates the Microsoft® digital inking technology (section 3.3.1) we do not distinguish between the two in our overview. (We dedicate chapter 3 to making clear this distinction.) We conclude the chapter in section 2.8 with a description of eight Tablet PC systems that have been studied and integrated into learning environments.

2.2 Didactic versus Constructivist Teaching

“Both educational research and educational philosophy generally hold that teaching beliefs and practices conform to one of two types of pedagogies, the didactic or the constructivist [194].” Wenglinsky compares the teaching styles of two early technology advocates, Seymour Papert and Patrick Suppes. In the 1960's Suppes developed and integrated computer-based programs as a means of drilling students to reinforce learning. This approach later became known as computer-assisted instruction (CAI). This type of computer

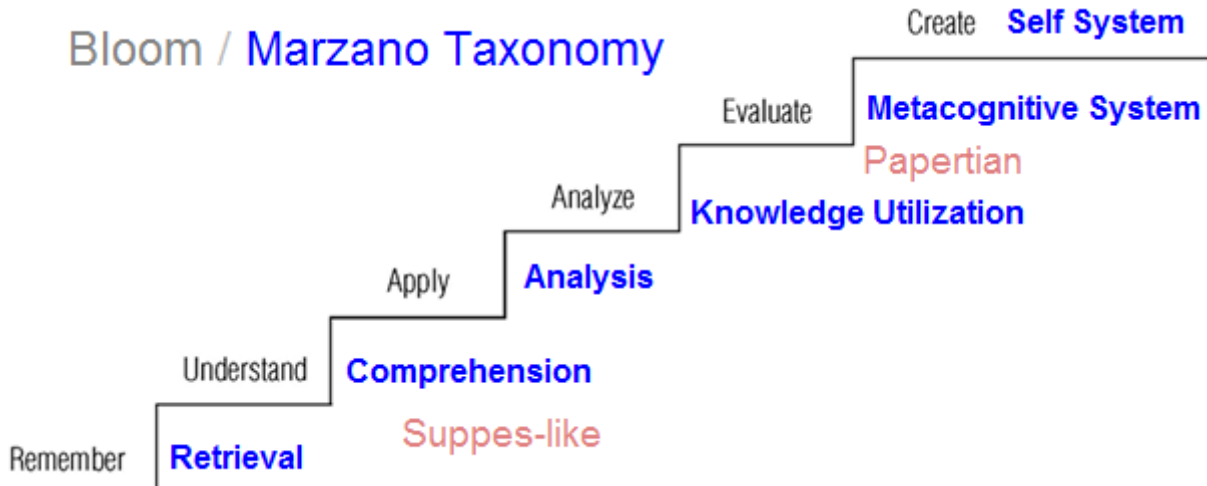


Figure 2.1: A comparison of the two models of teaching; the didactic characterized by lower level thinking skills taught in a teacher-centered classroom and the constructivist model where more student-centered instruction activities and higher-order thinking skills are possible.

assessment¹ of student knowledge has been criticized for failing to engage students in higher-order thinking. Papert, on the other hand, argued that students able to program a computer would learn how to think at a higher level of abstraction and could learn how to teach themselves [194].

Wenglinksy associates the didactic approach to teaching with Suppes and the constructivist approach with Papert. Over the past century, teacher-centered practices (didactic) have dominated while small variations of student-centered teaching (constructivist) have appeared in small, private, and elementary schools [43]. Teacher-centeredness is characterized by whole class instruction where teachers do almost all of the talking while students listen passively. Didactic refers to the traditional teaching model where students learn facts and basic skills as a group. In Marzano’s terms, this equates to the lowest level of information processing (Level 1: Retrieval) and in the Bloom taxonomy, Level 1: Remembering (figure 2.1).

See Hannafin [72] for an excellent discussion on the foundations and assumptions of technology-enhanced student-centered learning environments.

2.3 Computers in the Classroom

At the 1996 National Educational Summit meeting held at IBM, President Clinton addressed the attendees on the importance of academic standards, tests, and technology. The results of the summit are summed up by their issued statement: “We are convinced that technology, if applied thoughtfully and well-integrated

¹Assessment is the process of documenting, usually in measurable terms, knowledge, skills, attitudes and beliefs. Source http://en.wikipedia.org/wiki/Performance_evaluation

into a curriculum, can be utilized as a helpful tool to assist student learning, provide access to valuable information, and ensure a competitive edge for our workforce [117].” As a result, two billion dollars was made available for five-year grants from the Technology Literacy Challenge Fund [160]. With the money came a number of challenges. One challenge was to provide ubiquitous access to computers connecting classrooms to the outside world. This challenge was accomplished; money was paid and the hardware and network infrastructures were installed. Two challenges that have yet to be attained include the task of integrating educational software into the teaching curriculum (it should be as “engaging as the best video games”), and the requirement that teachers should be able to use the technology to teach [44, 199].

In *Oversold and Underused* [44], Cuban reports that techno-promoters across the board assumed that increased availability of computers in the classroom would lead to increased use and that increased use would lead to efficient and better learning. Three questions guided Cuban in his investigation and examination of these underlying assumptions:

1. In schools where computers are readily available, how do teachers and students use the machines in the classroom for instruction?
2. Have teaching and learning changed as a consequence of two decades of heavy promotion and investment in computers and other technologies? If so, what explains the changes? If not, what explains the stability?
3. Has the investment in computers and other technologies been worth the cost?

By addressing these questions Cuban presents a clear picture of how technology was integrated into the K-12 learning environment. To categorize the level of computer integration Cuban uses the following taxonomy:

1. Entry: as an instructor’s first exposure to technology.
2. Adoption: those instructors who allocate short periods of time to short activities such as how to start up an application, manipulate the mouse, and run applications.
3. Adaptation: classrooms allocating about a quarter of their time to using computers for homework and exercises.
4. Appropriation: teachers who regularly integrate technology into their lessons
5. Intervention: bold instructors who are willing to experiment with innovative ways of integrating technology into their learning environment. Such early adopters usually subscribe to the constructivists teaching style and integrate project-based instruction as a means of creating a more student-centered learning environment. (Craig Russell, an instructor in our fall 2007 and spring 2008 study, described in chapter 7, fits into this category.)

The answer to the question of how teachers and students use machines in the classroom for instruction is that elementary school teachers used computers as a means of enrichment or as a learning tool rather than as a daily activity. These enrichment activities included whole-group instruction, small-group work, and individual exploration. Based on extensive observations, interviews, and survey data, Cuban and his team

concluded that the integration of computers into classroom curricula and instruction was minimal; almost all teachers were considered to be at the entry and adoption level. The answer to the question of how, if at all, teaching practices were changed as a result of computer integration was that “teachers adapted an innovation to existing ways of teaching and learning that have dominated early childhood education for decades [44].”

Cuban generalizes his findings to conclude that across the country, most teachers and their students are nonusers or occasional-to-rare users of machines in classrooms. In the decade between 1990 and 2000, elementary teachers transitioned from nonusers to occasional users while computer use in high schools moved from occasional to serious use. High School use of computers was dominated by word processing programs, with small percentages characterized by drill and practice software, playing math games, running simulations, and, once in a while, demonstration of new concepts in math. Types of home use of computers reported by teachers in two Silicon Valley schools include personal use, preparation of school materials, email, searching the Internet, preparing tests and lesson plans, and recording grades. There are more recent qualitative studies such as [173] that characterize the overall attitudes of high school teachers regarding one-to-one computing, emphasizing the need for support during the early adoption phase.

How did teachers teach in K-12? Teachers lectured, facilitated group discussions, reviewed homework, worked on exercises, and occasionally used a VHS or overhead projector. Teacher-centered instruction was the norm. As always, the technological enthusiasts, or early adopters, represent an outlier of the norm [9]. Using nationally collected data these observations were confirmed: teachers lectured and students listened, read textbooks, and completed individual exercises. In [135], Norris et al. argue that “the reason there has not been an impact of technology [in K-12 institutions] is that students have actually, for all intents and purposes, not used the technology. And, the reason for this non-use lies not at the feet of the teachers, but rather in the very real lack of access to technology.” This sentiment is echoed by Becke [13]. This statement supports the assumption that access to technology leads to computer use. Unfortunately, one cannot investigate whether teaching has been at all affected by the use of technology if the technology is not used. There are more recent publications that explore teachers’ changes in patterns of computer usage [91, 152], and the role of differences in beliefs about technology integration and student-centered instruction practices [82, 197].

At the university level faculty and students use computers extensively to research, write (word processing), communicate (email), search the web, and prepare for courses. However, there is little to no use of these technologies for direct classroom instruction, aside from the small number of classes taught exclusively in computer labs. Some accredit the lack of computer integration into university classrooms to the inherent

pressures placed upon professors to earn tenure through research publications without reward for innovative teaching practices [36]. One popular initiative that began in 1996 at Georgia Institute of Technology was the Classroom 2000 Project which helped provide ubiquitous access to computer technology by automating the capture and access of in-class instruction (section 2.5). Computer technology was seamlessly integrated into the teaching and students had the ability and motivation to access the archived class session outside the class.

2.4 Problems with Conventional Technology

The Computer Science Department at the University of Illinois uses much conventional information technology to enhance learning and offer distance learning degrees and courses. The department has used web and video streaming technologies for asynchronous education and distance-learning since 1995 and used bulletin boards, news, and on-line testing to enhance its courses. The University of Illinois has been an early adopter of computer technology for education.

Starting in 1959, before the advent of the web, the PLATO system [201] was used to deliver instruction on campus. Online chat and bulletin-board notes features were added in the early 1970s. In 1975, Control Data Corporation (CDC) established PLATO-IV [176] as a commercial educational product that, by 1985, had established systems in over 100 campuses around the globe. Later versions of the notes system became popular for providing a means for students to collaborate with one another electronically, such as Lotus Notes (see <http://www-128.ibm.com/developerworks/lotus/library/l5-NDHistory/>), a system created by alumnus Ray Ozzie [159]. The PLATO system is an example of using a client-server computer technology for education. The client-server approach is used in most current web-based educational communication and interaction systems.

Modern electronic web, chat, newsgroups and email systems are examples of distributed educational information technologies that have their origins in a workstation/mainframe perspective of computing. These technologies allows local and remote students to communicate outside of fixed office hours and provides both synchronous and asynchronous forms of distance learning.

However, the recent arrival of ubiquitous computing introduces a non-invasive, “natural”, notion of human-computer interaction. New input and output devices encourage communication based on handwriting, speech, and gestures. Computers have become small and mobile and the advent of the Tablet PC offers a familiar modality of input that was long ago replaced by the mouse and keyboard [133]—the pen and paper modality. As early as 1998, the problems with keyboard and mouse input in a wireless classroom were



Figure 2.2: eClass learning environment. A) Projector of either internet browser or whiteboard contents. B) Projection of previous lecture slides C) The instructor annotates a PowerPoint presentation or writes on a blank whiteboard. The ceiling and walls provide small microphones and cameras for recording the context of the classroom experience.

documented [69].

2.5 eClass (Formerly Classroom 2000)

eClass pioneered the capture and access of classroom lectures by providing a detailed record (context) of college lectures in the form of autonomously generated media-enhanced notes. The augmentation of traditional notes with multimedia such as audio and video provided for a more detailed record than was previously possible with pen and paper. How an instructor delivered a lecture was little changed and overall the setting was moderately unobtrusive. The recording devices were carefully positioned in the ceiling to reduce the overall “awareness” of observation (figure 2.2).

Unique to the eClass project was that they were able to study the impact of their system over the course of a three-year study at multiple institutions in a large number of courses. The results presented from their longitudinal study [86] illustrated a number of interesting points:

- eClass did not have a negative impact on attendance.
- eClass did not have a measurable impact on performance (based on grades).

- The online notes generated from automated capture of college lectures were appreciated and used, and media augmentation was also appreciated.
- The captured notes were mostly used to review lectures shortly after they occurred and for exam cramming.

Lessons learned from eClass have inspired the development of our system over the years. The requirement that the integration of the technology needs to be seamless is challenging, especially when considering the introduction of new technologies that support enhanced pedagogical practices.

It is difficult to evaluate whether the failure of eClass to have a measurable impact on student performance was a result of the requirement that the capture of lecture materials require no extra instructor effort. The initial design of the system was based on the automatic capture and archiving of large class lectures for later access. When Abowd and Brotherton published their reflections and “lessons learned” [86], they made a number of recommendations that seem fanciful, if the initial requirement of automating the capture holds. Two such requirements are an improvement in the quality of the captured content, and that the system should support “collaborative” note taking. Collaborative note taking such as that discussed in [17, 126] makes it even more difficult to provide a seamless integration. While the level of detail provided of an instructor’s lecture was seemingly maximized in the eClass system, the inking technology of the Tablet PCs has the potential to offer a greater level of detail such as in their recommendation that integration needs to be smarter. Figure 2.3 illustrates an example of their automated archiving of a class session.

For a broader synopsis of computer-enabled classrooms and computer-integrated classrooms (CiC) see [78, 144]. For an overview of Computer Based Instruction (CIB) see [74, 111].

2.6 Tablet-based Instruction and Research

Typically, pen-based systems (not exclusively Tablet PCs) have been studied for their ability to perform a variety of tasks, from information search and retrieval to educational applications. More specifically, they have been studied for their effectiveness at performing annotation and note-taking [11, 34, 46, 47, 61, 84, 90, 126, 136, 142, 165, 195], handwriting recognition [83, 96, 150, 175, 182], sketch book design [32, 153, 162, 196], sketch recognition [157], gesturing [76, 77, 115, 138, 155, 204], drawing and illustration [32, 56, 63, 87, 141], information acquisition [62, 120], classroom interaction [27, 33, 69, 89, 98, 107, 130, 137, 140, 168, 196], presentation [18, 69, 80, 99, 113, 148, 185, 195], and delivery of instruction [15, 28, 67, 68, 79, 80, 86, 88, 97, 106, 126, 172].

Today’s Tablet PC’s are significantly different from pen-based systems of the past. The ability to use a pen to directly write on the surface of a computer screen removes the frustrating barrier of old input devices

eClass Automated Lecture Archive

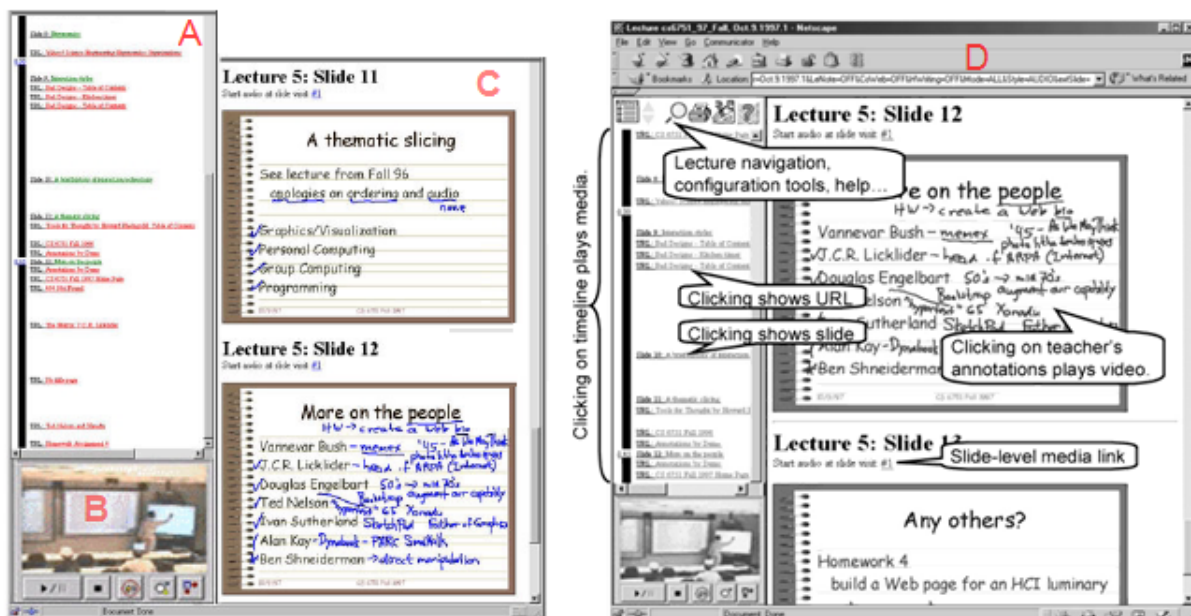


Figure 2.3: An example of an eClass automated archival of a class session. A) Generated time line includes green links (web URLs), red links (instructor slides) and blank links (video excerpts). B) A recording of the live video feed is displayed in the lower left hand corner and C) the set of post-lecture annotated instructor slides. Clicking on the blue annotations links to indices created in the video feed. D) Snapshot of notes taken by eClass presented in [86]

where users were forced to move their hands holding a digitizer on top of a pad on the side while trying to focus their eyes on the screen to see the outcome. One very important distinction among Tablet PCs manufactured today is the difference between the passive digitizer display and active digitizer display [53] (see section 3.3.1). Active displays are incorporated into the newest Tablet PCs and use a radio frequency-based pen and screen digitizer to provide a smoothness of flowing ink as it appears on the screen. Digital ink has been transformed into what Microsoft calls “rich inking” [85].

Several early studies have shown that the Tablet PCs serve as useful classroom presentation tools. Each of the studies conducted using pen-based custom classroom software report success at fostering interaction in the classroom environment[6, 15, 21, 28, 37, 58, 92, 129, 148, 149, 168, 184, 190, 200]. Additional studies have involved custom classroom software: UW Classroom Presenter [6, 18], Ubiquitous Presenter [195] and Classroom Learning Partner [107], (both extensions of Anderson’s Classroom Presenter system); DyKnow, a commercial product developed by David Berque, Professor of Computer Science at Depaw University [15]; MIT’s InkBoard [196]; Livenotes [81, 126] from University of California at Berkeley; ReMarkable Texts [183] and ChempPad [179] from Brown University; and OrganicPad [139] from Clemson University.

2.7 Tablet PC Systems

There are a number of Tablet PC applications currently in use that were not exclusively designed as instructional tools. These include Microsoft’s Journal, OneNote, and PowerPoint with the integrated inking functionalities. Before classroom Tablet PC systems were built and freely distributed, early adopters of Tablet PCs attempted to integrate these applications into their classrooms, because they were really the only applications available. As a case in point, at the University of Illinois at Urbana-Champaign we met a professor in the School of Business who was extolled for his teaching ability and popular for integrating technology into his lectures. For years, this instructor came to class with a “prepared” blank PowerPoint presentation which for all intents and purposes was an ad hoc implementation of a very limited electronic whiteboard. This example illustrates not only the need for software designed for instructional purposes but how the lack of applications forces users to settle for what currently exists. In this case, the absence of a suitable application led an instructor to prepare a blank presentation in order to make a presentation. It worked reasonably well.

As Tablet PCs became more pervasive in academia, in part as a result of generous grants awarded by Hewlett Packard in the form of Tablet PC equipped classrooms, and Microsoft Research funded Tablet PC applications initially intended to work with the Learning Experience Project [12] (ConferenceXP Architec-

ture), more open source Tablet PC applications have become available. The release of the 1.7 Tablet PC SDK provided a vehicle for building applications designed exclusively for the Tablet PC. In the next section we summarize a number of the systems designed for instruction and classroom use.

2.8 Overview of Tablet PC Systems for Education

In this section we will present eight systems designed for use with the Tablet PC. We first present UW Classroom Presenter developed at the University of Washington. Five of the remaining systems were developed at other institutions, but were built off of the Classroom Presenter code base: Ubiquitous Presenter and Note Blogger developed at the University of California at San Diego, and Classroom Learning Partner (CLP), Group Learning Partner, and Elementary Learning Partner developed by the CLP group under the supervision of Kimberle Koile at Massachusetts Institute of Technology. The remaining three systems are LiveNotes developed at the University of California at Berkeley, a commercial product called DyKnow, originally developed at Depauw University in Indiana, and the MIT Inkboard.

2.8.1 UW Classroom Presenter

UW Presenter is now in its third major release—Classroom Presenter 3—and is now integrated with ConferenceXP 4.1, also under development at the Center for Collaborative Technologies at the University of Washington (<http://cct.cs.washington.edu/>). Professor Richard Anderson has led the research and development of UW Classroom Presenter from its inception and over the years has investigated how the Tablet PC technology can facilitate a learning environment conducive to active learning [7, 8, 114, 147–149, 168, 190].

Classroom Presenter (CP) is an interactive Tablet PC system for delivering instructor slides (called decks in CP terminology) and sharing of digital ink with student participants. One example of sharing digital ink between students and instructor is that the system allows students to submit portions of their whiteboards anonymously for instructor review. Figure 2.4 provides an illustration of UW Classroom Presenter 3. A number of systems (presented below) have been developed by extending the UW Classroom Presenter. There is even a version in development for the XO laptop initiative (one laptop per child) (<http://xo.orderedpixels.com/>).

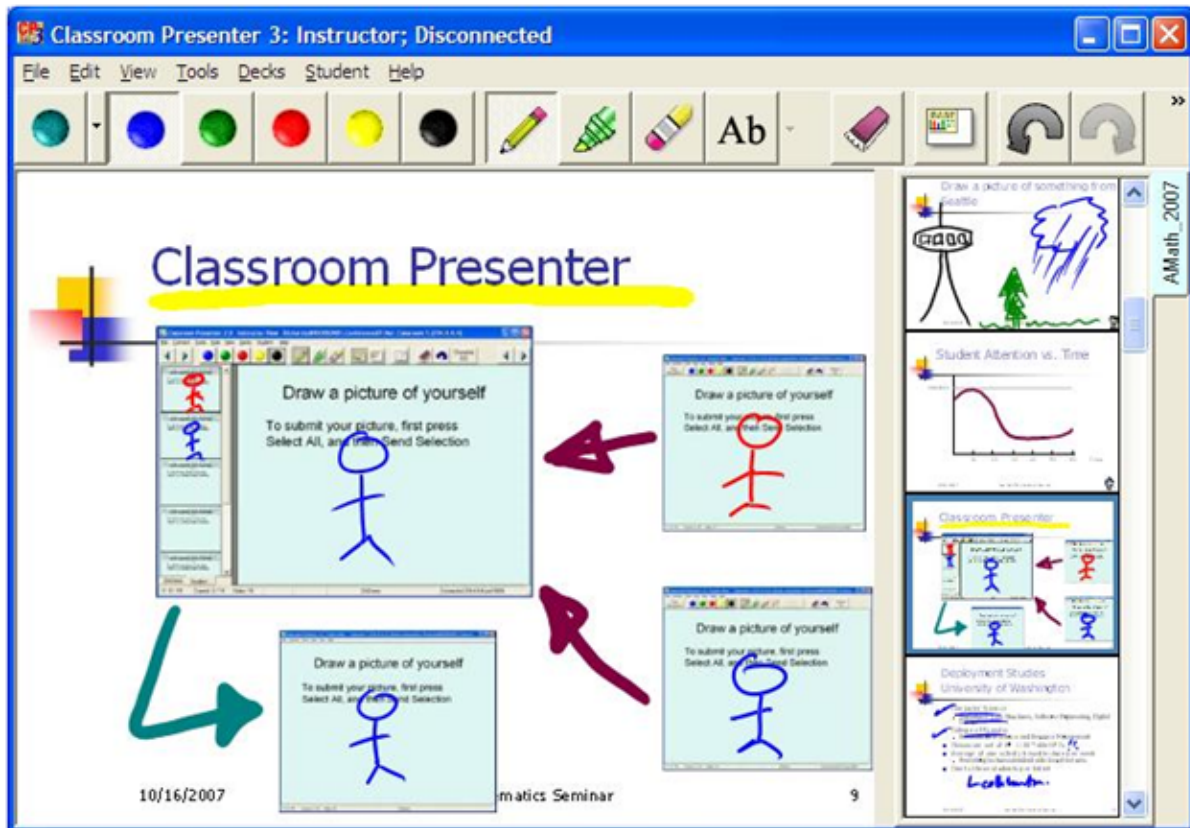


Figure 2.4: UW Classroom Presenter 3. A Tablet PC interactive whiteboard system supporting the importing of PowerPoint presentations and transmission of instructor-created content and student created anonymous submissions.

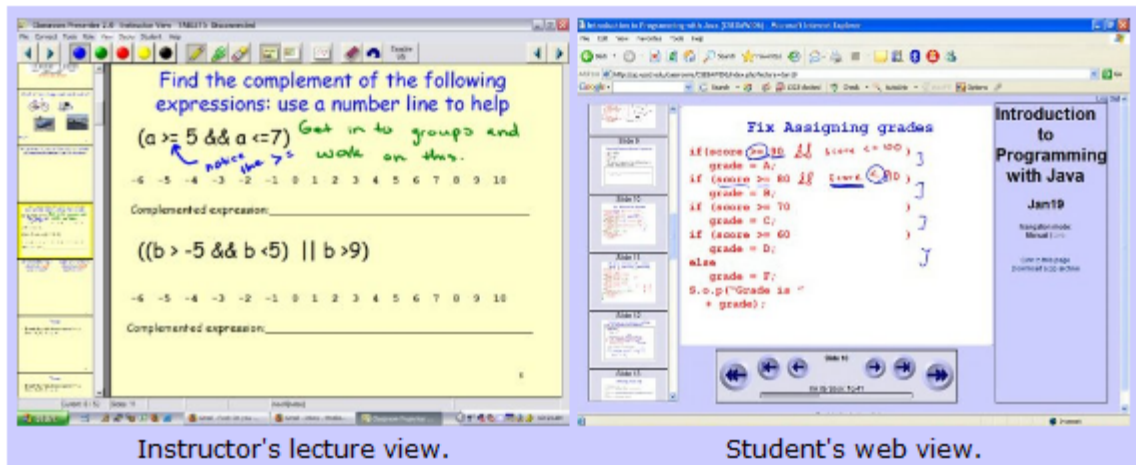


Figure 2.5: Ubiquitous Presenter, also based on the UW Classroom Presenter project provides a proxy server by which students can access the web to view instructor slide content in real-time and also participate in polling (see part B of figure 2.6). (source: <http://up.ucsd.edu/about/WhatIsUP.html>)

Ubiquitous Presenter

Ubiquitous Presenter (UP) was developed at the University of California at San Diego and is an extension of UW Classroom Presenter [195]. The focus in this project is to support access and replay of lecture content on the web (similar to eClass). One difference between UP and other systems is that students can synchronously participate in and out of class via a web-based interface. In this way, the implementation of their system provides for ubiquitous access anywhere, and on any device, without the requirements of special software installation. Figure 2.5 illustrates the Tablet PC interface from the instructor's perspective and the web-based interface for non-Tablet PC users.

The Java implementation provides non-Tablet PC users with the ability to ink, provide textual input, and press buttons to select and submit poll responses (figure 2.6).

NoteBlogger

Another application and extension of UP is their more recent system called NoteBlogger (NB) where they investigate the effects of student blogging (sharing of notes) taken during the course of a lecture [17]. NoteBlogger (NB) is a Tablet PC application that provides selected students with the ability to add personal ink annotations to instructor created presentations in real-time. The content is posted on the website with the instructor's annotated slides allowing for public shared access (figure 2.7). Note the addition of the student and course staff annotations to the web-based java implementation illustrated in figure 2.6.

Similar to LiveNotes (section 2.8.2) the research question is whether opening up additional channels of communication to more students will help more students get involved and engaged in the lecture (a known

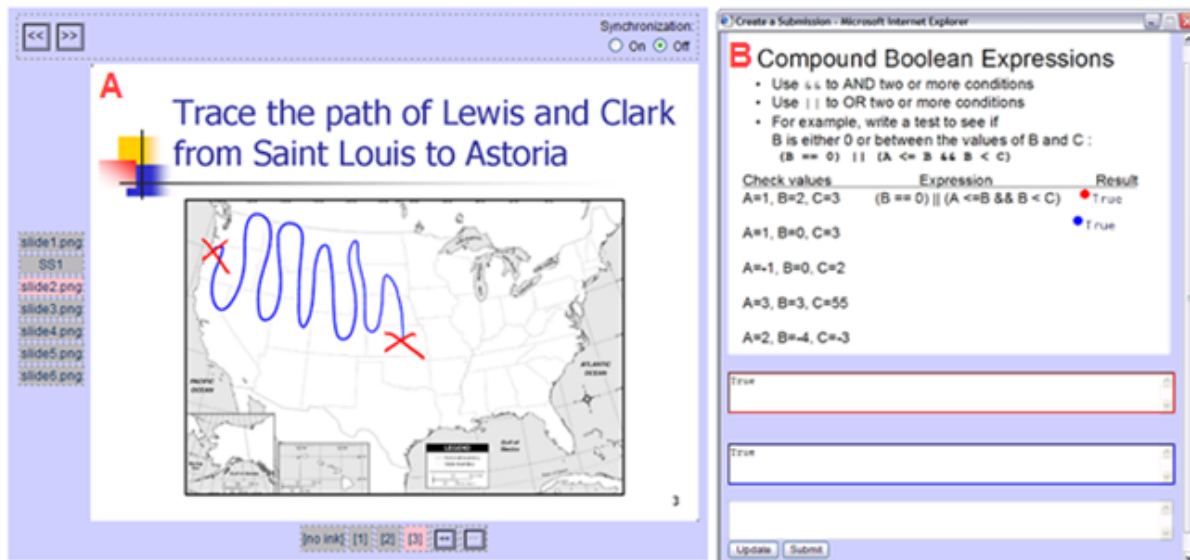


Figure 2.6: A) Ubiquitous Presenter interface for non-Tablet PC devices B) Ubiquitous Presenter provides a browser-based portal to participate anonymously in assessments in the form of polls sent by the instructor to the students in the class.

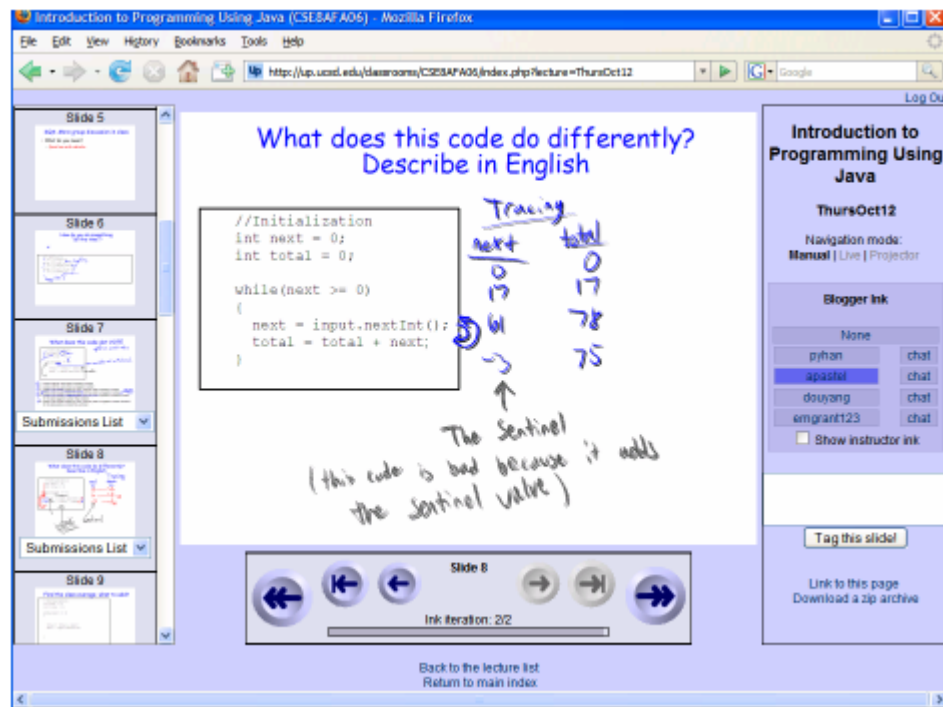


Figure 2.7: Web-based interface for viewing student blogger's ink annotations

problem in large lectures [7, 55, 174]).

Classroom Learning Partner

The focus of the Classroom Learning Project (CLP) is to increase instructor-student interaction in support of increased student learning. The CLP research group is directed by Dr. Kimberle Koile, a Research Scientist at the MIT Center for Educational Computing Initiatives. The CLP group is investigating the hypothesis that the use of assessment and feedback systems where students can submit ink responses to polls and exercises in large lectures supports student learning by increasing student attentiveness in class. Note that their system is not limited to the use of force-response type questions such as multiple choice questions. The CLP group is continuing to investigate new in-class exercise types, including sketched answers, and associated interpretation and aggregation routines [107].

Built on top of the UW Classroom Presenter system, CLP provides an infrastructure to support in-class exercises in large classes. Their solution to the overwhelming amount of student feedback received upon student submissions is to aggregate these responses into a number of equivalence classes by comparing those answers to instructor-specified correct answers and incorrect answers, and/or by clustering student answers [107, 108]. The idea is to present a simplified summary of the distribution of student answers in the form of a histogram and example answers (figure 2.8). This is a problem that we address in chapter 7 of this thesis.

Group Learning Partner

Group Learning Project (GLP) is an extension of CLP that explores how the Classroom Learning Partner can foster group interaction in large lectures [108]. The application supports the sharing of ink strokes between student-assigned groups. The instructor can assign students to groups based on four different criteria: by students' geographic locations in the classroom, students' names or machine names, randomly, or by allowing students to select groups themselves. Additionally, the instructor has the ability to monitor group work spaces to view a group's collaborative work (not assigned assessments) and assigned exercises (figure 2.9).

Elementary Learning Partner

Elementary Learning Partner (ELP) is another extension of CLP that integrates a whiteboard application for use in elementary schools [107]. The CLP development team wants to investigate whether the success of CLP in the undergraduate classes at the university level can be extended to the K-12 environment. (We ask a similar question and present our results in chapter 7.) A number of modifications were made to the CLP

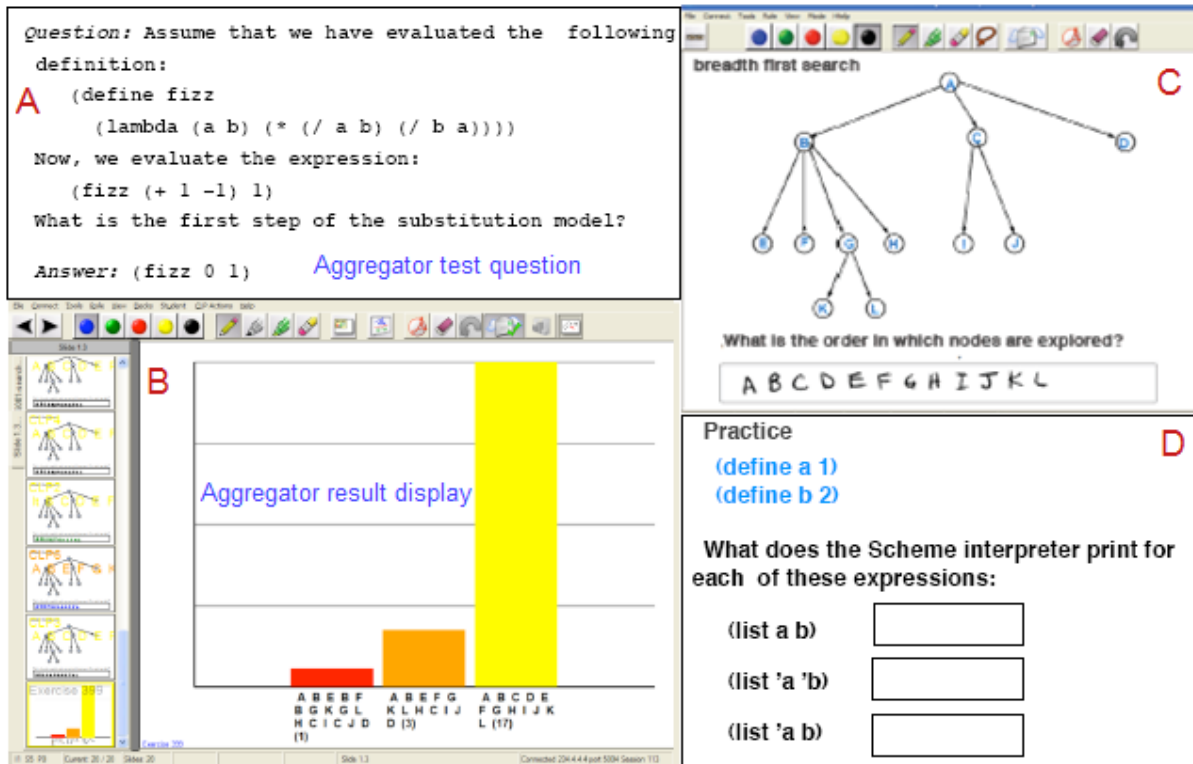


Figure 2.8: A) Example question sent to students Tablet PC devices B) Aggregate of results displayed on instructor's display of question presented in part C) representing the Tablet PC interface. D) Instructor Authoring Tool (AIT) allows an instructor to add exercise information to their PowerPoint presentations including the location of a student's ink answer (the answer box), and the expected type for that answer such as an integer or character.

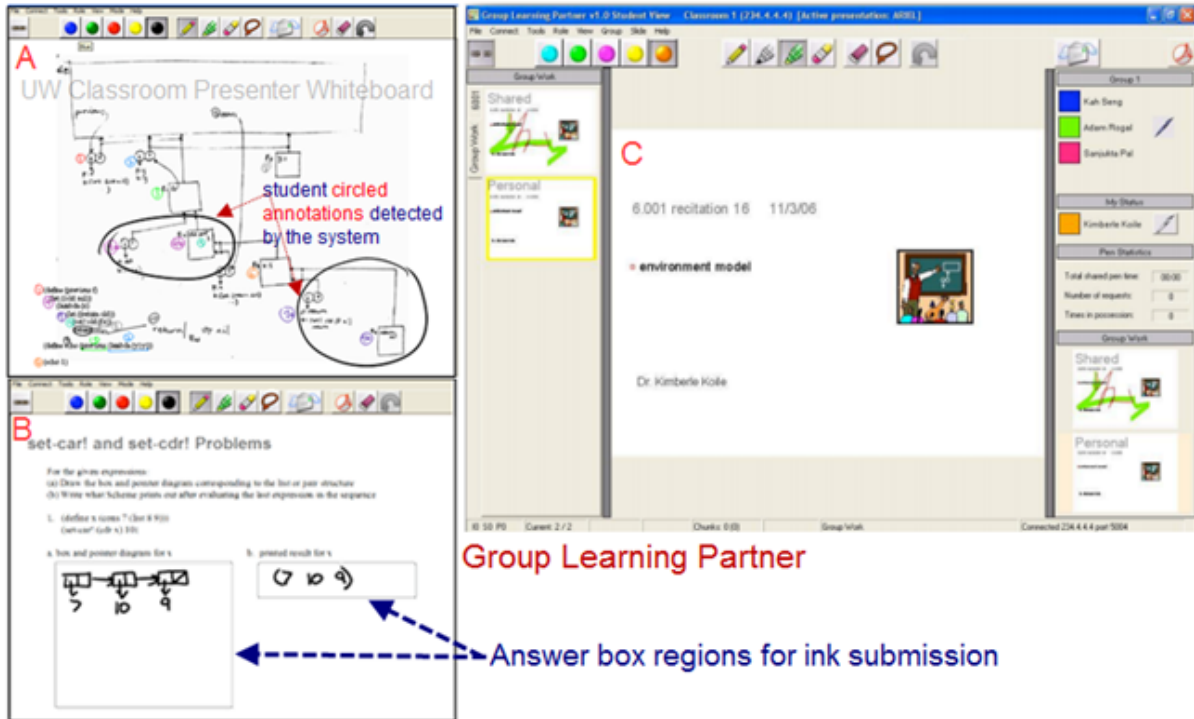


Figure 2.9: Group Learning Project (GLP) is an offshoot of Classroom Learning Project (CLP) which integrates collaborative exercises to facilitate group work. The system integrates an advanced ink analysis engine to decipher annotations for correct identification of diagram equivalence of code (A). B) Answer box regions are used for submitting specific regions of a student's whiteboard. C) The interface for the GLP that allows students to collectively contribute to solving exercises assigned by the instructor. Student's can switch between viewing the instructor's presentation slides, their personal slides, and the group's slides. Writing on the group slide is limited to one at a time to eliminate the problem of ink "collisions".

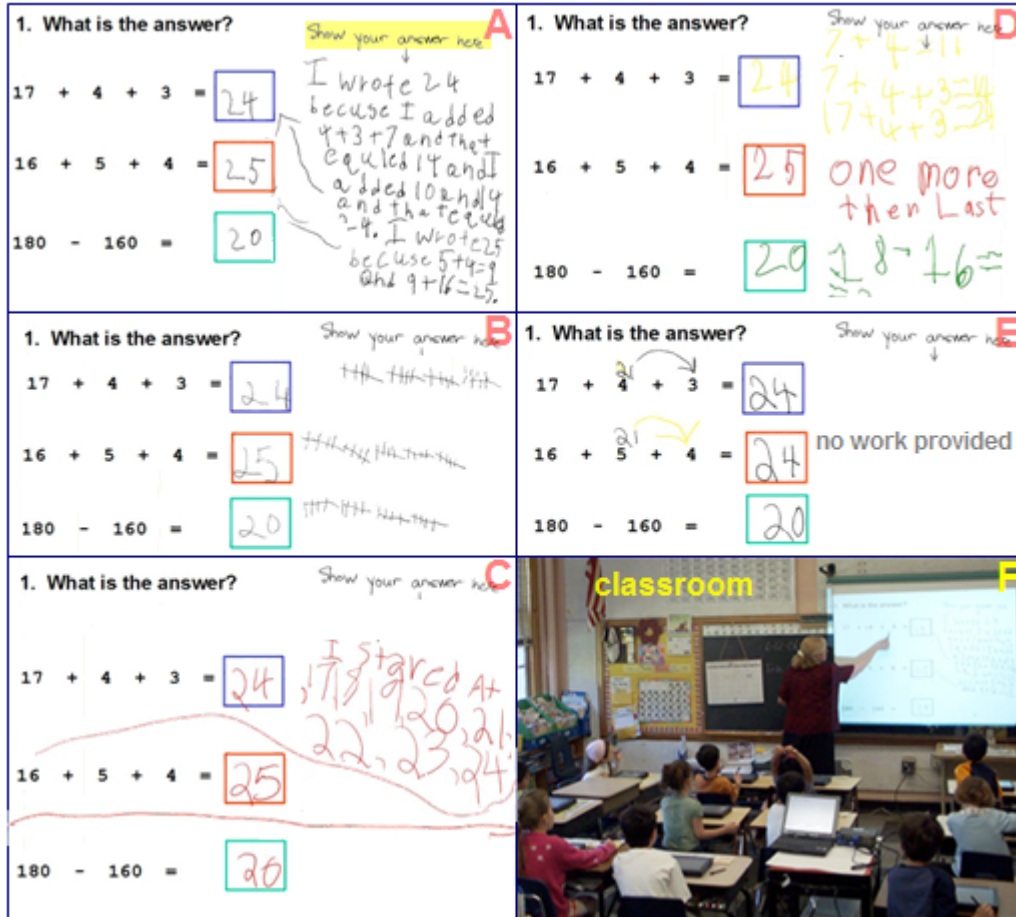


Figure 2.10: Elementary Classroom Partner is an extension of the Classroom Learning Partner (CLP). The illustrations in A-E represent actual student submissions to three arithmetic exercises. The highlighted portion in A (our emphasis) was written by the instructor and says “Show your work here” with an arrow drawn below the workspace. F) A snapshot of the classroom using ELP system.

system to better integrate the application into an elementary school setting such as the addition of more colors and a simplified user interface. Students submitted ink solutions to exercises and an instructor was able to assess and display student work (figure 2.10).

2.8.2 LiveNotes

LiveNotes was developed at University of California at Berkeley as a tool for cooperative small-group learning via collaborative note-taking. The project addresses the problem that large classes pose on student attention, engagement, and retention. A few assumptions are made that guided the development team; first, that learning is more effective in small groups, secondly, that student interaction is enhanced through communication with his or her peers and that through this interaction students are able to explain the material to one another. The research question is whether or not technology can be used to foster small-group

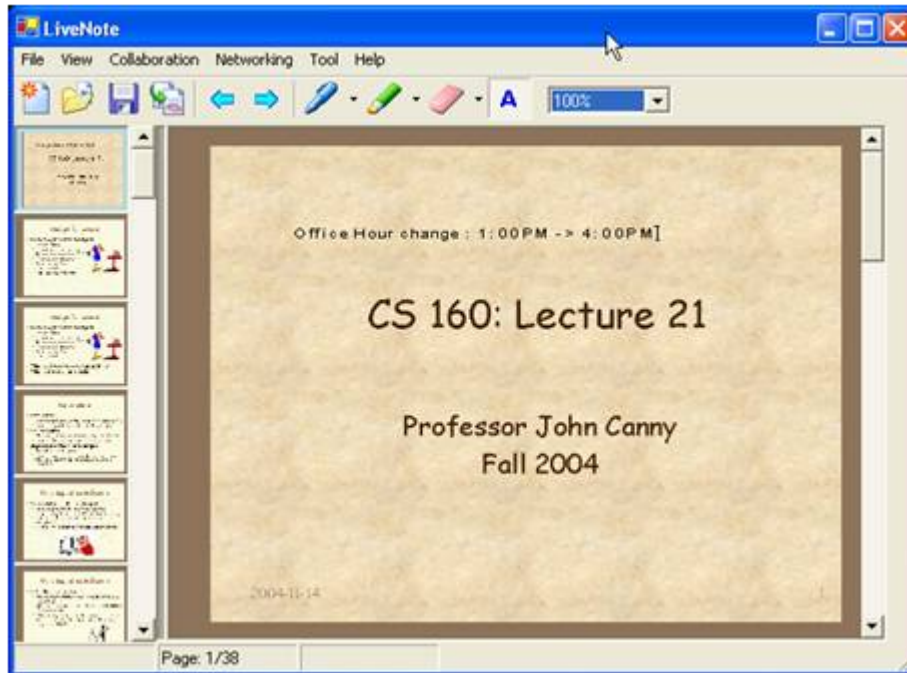


Figure 2.11: LiveNotes implementation in C# for the Tablet PC. This is the interface used by students and instructors on Tablet PCs.

learning in large classrooms. In a LiveNotes class session, students are provided with a skeletal form of the instructor's PowerPoint presentation (figure 2.11).

Students with Tablet PC devices can annotate with the stylus and students with laptops can contribute to the notes through the keyboard on their devices (figure 2.12).

Feedback in the form of polls is also available for students with access to the internet. Figure 2.13 illustrates the lecture feedback dialogue and the instructor's display of aggregated results.

2.8.3 DyKnow

DyKnow® (figure 2.14) (<http://www.dyknow.com>) is currently the only commercial Tablet PC-based classroom learning system on the U.S. market. The DyKnow system provides the sharing of instructor presentations (similar to the systems mentioned above) and provides a number of additional features well suited for the K-12 environment. DyKnow Vision® provides teachers with the ability to monitor student work, retrieve student panels (whiteboard contents), and display and replay student work. Students can interact with the Tablet PC to participate in classroom activities, collaborate with other students, and access their classroom portfolios from any internet connection. DyKnow also provides the administrative classroom management functionalities available in products like NetSupport School (<http://www.netsupportschool.com/>), Net Control 2 (<http://www.netcontrol2.com/>), and LanSchool (<http://www.lanschool.com/>).

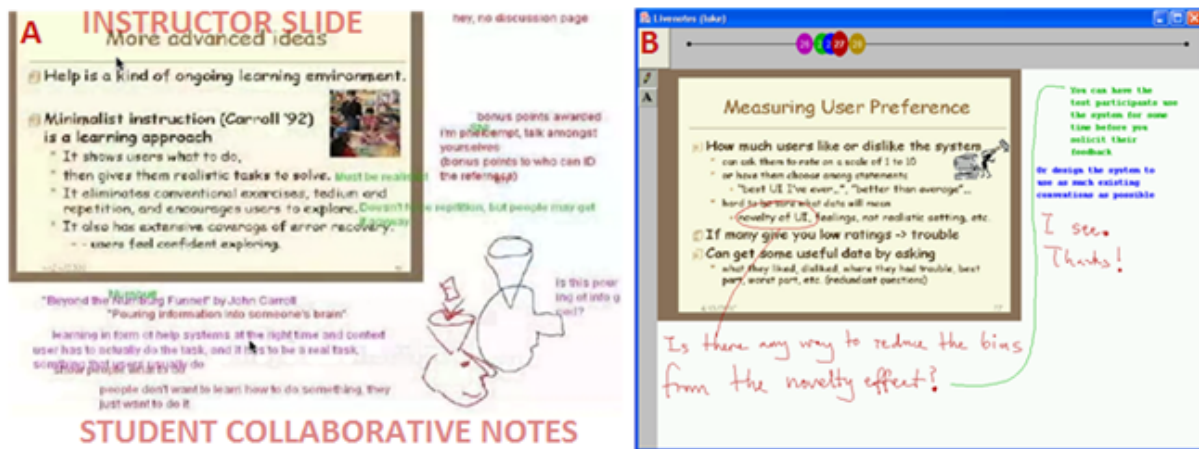


Figure 2.12: A) C# LiveNotes interface for collaborative note taking. The different colored annotations (both textual and diagrammatic) are contributions from selected students in a class session. B) Java interface of LiveNotes (an earlier release) for collaborative note taking for a single slide. The colored numeric circles represent the student contributions; their placement on the timeline representative of the “begin” time stamp for the annotation, the color and number unique to the student entry. This screenshot of LiveNotes was taken from a video posted on sourceforge

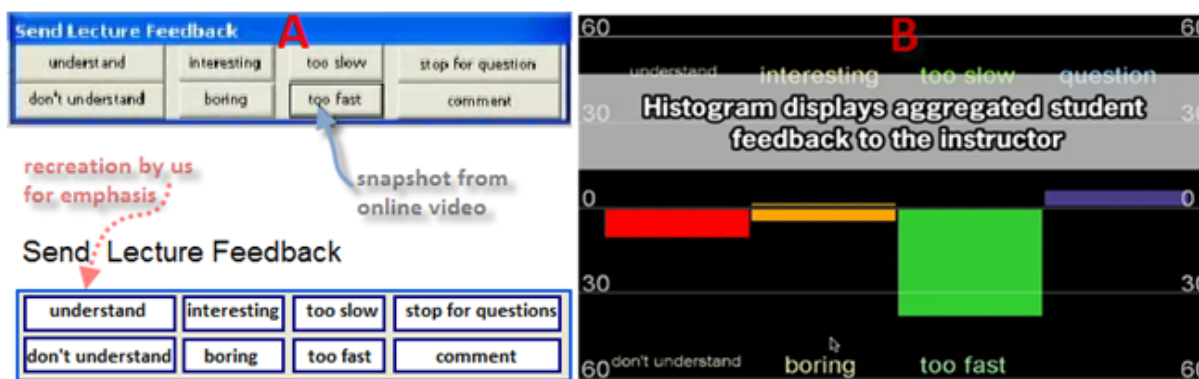


Figure 2.13: A) LiveNotes user feedback polling dialogue B) histogram of aggregated user feedback displayed on the instructor's machine

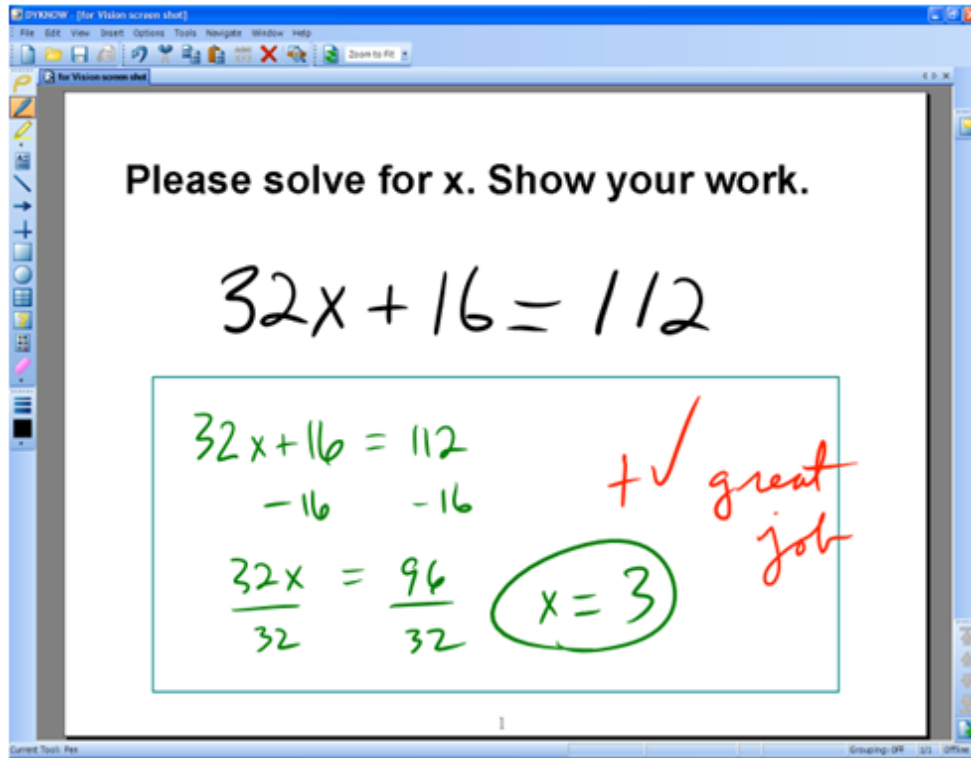


Figure 2.14: DyKnow Vision. (Source: <http://www.dyknow.com>)

2.8.4 MIT InkBoard

The MIT InkBoard (figure 2.15) was one of the earlier and most impressive academic collaborative sketching systems. The developers designed an outstanding user interface and their system architecture supported the integration of the Microsoft Conference XP system. ConferenceXP supported the audio and video functionalities which functioned similarly to an instant messenger program with the addition of the inking technology. The system used Microsoft SQL server to store “studios” which housed different drawing / collaborative sessions. One extremely powerful function of the system was the ability to zoom in and out on the canvas creating an essentially unlimited whiteboard. We experimented with the system in one of our earlier pilot studies where the instructor used the application as a presentation tool.

The most impressive feature of the system was the integration of a timeline (figure 2.16) feature for reviewing the history of the ink board events. The user interface was similar to the graphical interface used in video editing systems. Using a sliding bar mechanism a user can navigate the history timeline; ink strokes not yet created are displayed as translucent ink strokes. Selecting and deselecting user icons in the buddy list allowed a user to hide that user’s ink strokes.

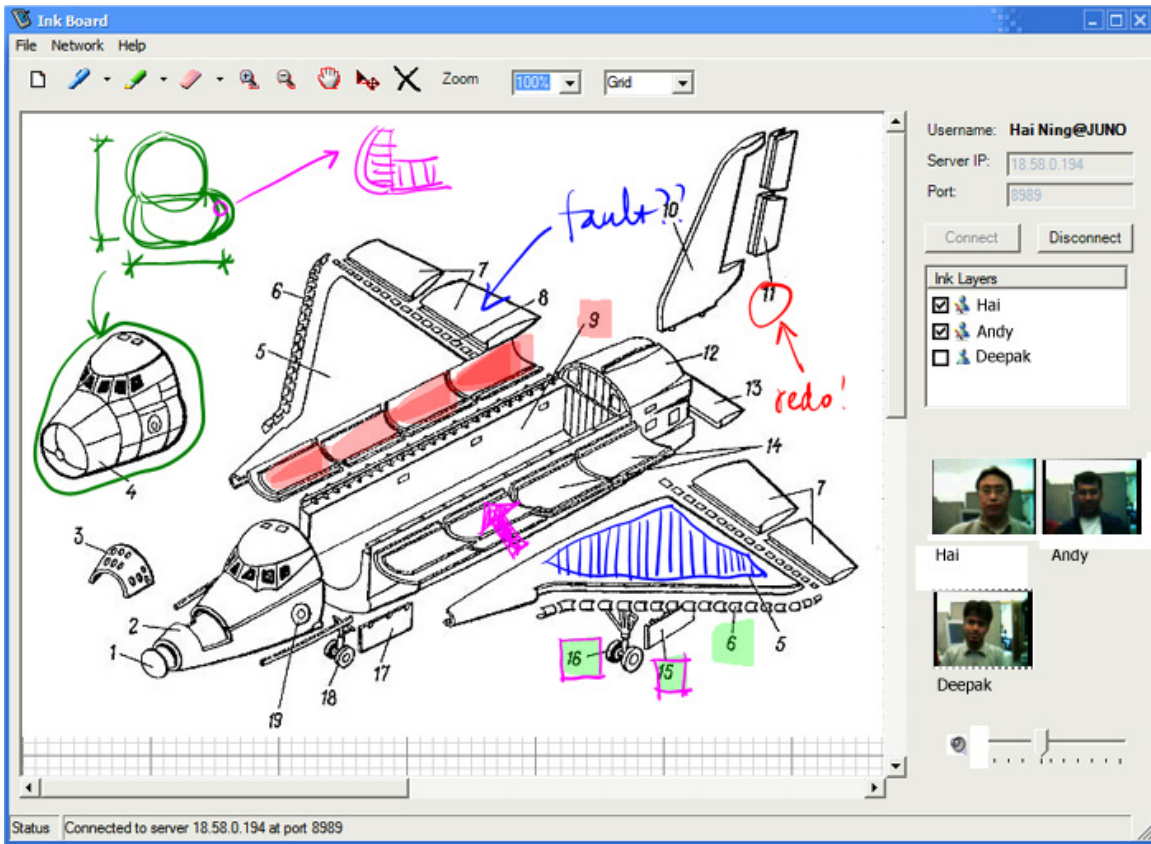


Figure 2.15: MIT Inkboard interface. In this example an image has been inserted over the grid background which is the default canvas appearance

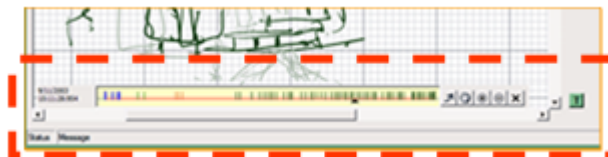


Figure 2.16: Ink board time line feature.

Chapter 3

The Potential for Tablet PCs

Microsoft's Tablet PC, the first Dynabook-like computer good enough to criticize

Alan Curtis Kay,

<http://en.wikipedia.org/wiki/Dynabook>

3.1 Introduction

Over the past three decades, educational institutions have attempted to improve the classroom experience by integrating numerous technologies. The ROI (return on investment) promised by the investment of technology in education has repeatedly fallen short of its goal. Study after study report how the “student experience” has been enriched, but seldom provide statistical evidence to support their promise of improved learning. Laptop and computer workstations have yet to be integrated into a teaching curriculum with even a modicum of success [44].

We believe that the Tablet PC provides our best means for integrating technology into learning environments because it is the closest parallel to pen and paper, which we know, as a modality, has been integrated into learning environments with success.

In section 3.2 we provide a brief history of the Tablet PC as it relates to the pen-input devices of the past. In section 3.3 we distinguish the Tablet PC from its predecessors by explaining the differences between types of digitizers (section 3.3.1), form factors (section 3.3.2), and inking technologies (section 3.3.3). We also explain the similarity between writing with pen-and-paper and writing on a Tablet PC (section 3.3.4). In section 3.4 we debunk a number of “common myths” about the Tablet PC. In section 3.5 we illustrate the potential for Tablet PCs through an exploration of activities related to note-taking and in section 3.6 we present examples of instructor use. The final section, section 3.7, describes an interesting demonstration that we did in 2004 at the opening ceremonies for the new Computer Science building, the Siebel Center. We could not give surveys or exams, so we could not do a formal “study,” but the demonstration illustrates a number of the points made earlier in this chapter.

3.2 A Brief History

Many people do not know that Allan Kay conceived of the tablet mobile computing device well before the first commercial product, GRiDpad (see below) surfaced. See Maxwell's 2006 dissertation detailing the history of the Dynabook [127].

...the Dynabook concept which defined the conceptual basics for laptop and tablet computers and E-books, and is the architect of the modern overlapping windowing graphical user interface (GUI). Because the Dynabook was conceived as an educational platform, Kay is considered to be one of the first researchers into mobile learning, and indeed, many features of the Dynabook concept have been adopted in the design of the One Laptop Per Child educational platform, with which Kay is actively involved (http://en.wikipedia.org/wiki/Alan_Kay).

The history of electronic pen-based computing dates back to the late 1980's when Grid Systems Corporation introduced the GRiDpad, soon to be followed with the release of two similar products, GO's PenPoint [25] in 1991 and Apple's Newton [128] in 1993. Poquet Computer Corporation, founded in Santa Clara, California, introduced its PoquetPad for "walking workers." At the time it cost \$1,995, weighed only 1.2 pounds, and ran on two AA batteries for up to 16 hours. Microsoft, in 1992, entered the market with software titled Microsoft Windows for Pen Computing, which contained an API for developers of pen-based applications. Unfortunately for these early pioneers, pen-based computing did not really take off until 1996, with the design of 3Com's Palm Pilot. The palm was a huge success, likely due to the focus on pen-friendly applications, its convenient size, and simplified user interface. An onslaught would follow with an array of new pen tablets and smaller handheld devices, many of which would make use of Microsoft's Windows CE or Pocket PC operating systems and a passive digitizer display (section 3.3.1). In 2002, Microsoft introduced what they called the Tablet PC—the next generation mobile personal computing system.

The essential feature of early pen-based computers was the incorporation of digital inking into applications. Digital ink uses a stylus (a virtual pen) as input and can store ink or recognized text. A group of researchers at the Institute for Complex Engineering Systems at Carnegie Mellon University were pioneering digital inking when they developed the following definition in 1998:

Digital Ink is a design research concept. Part design, part critique, it is the integration of current and future technologies into a mobile and socially familiar object. Digital Ink is a sophisticated pen that allows people to take notes, sketch, and save the raw data they generate, digitally and automatically. It strives to turn mobile computing and interaction on its head by turning the monitor into a piece of paper and the keyboard and mouse into the pen itself. It's designed so

people can do things they normally do with any pen, but also fax, print, plan and correspond with others [95].

3.3 Definition and Distinction

Hulls and Theys [80, 180] do a nice job introducing the Tablet PC and providing descriptions of use in educational settings for new users of Tablet PCs.

3.3.1 Active and passive digitizer

There are two types of digitizers integrated into pen-based devices, *active* digitizers and *passive* digitizers. While both of these technologies detect, collect, and convert analog data (generated from user input) into digital information, the active digitizer is the technology used in Tablet PC devices¹. The active digitizer provides higher resolution, higher accuracy, and overall superior performance when compared to the passive digitizer. “Higher resolution” of the active display refers to the faster sampling rate (the number of samples of data collected by the digitizer per second) of the active digitizers. “Active RF” refers to the fact that the pen contains a miniature RF component that transmits through the computer screen to an antenna positioned behind the LCD. A controller chip in the digitizer takes samples of the pen’s position about 133 times a second. This is a huge improvement over the traditional mouse device which typically takes samples of its position 30-40 per second. As a result, the active digitizers provides a great deal of information for each ink stroke² including pressure-sensitive information which adds the natural smooth feeling of real strokes of ink with varying width. Thus, users are now able to use the Tablet PC just as if they are using pen and paper. With the advent of sophisticated inking technologies and computationally powerful systems, it now seems reasonable that Tablet PCs will become useful instructional delivery and learning tools. Teachers and students can illustrate well, write in a “natural” hand writing manner, and erase and edit their annotations with ease.

There are a number of differences between the passive digitizer (also called resistive), the older technology, and the active digitizer, the more recent. Active digitizers require the use of a special stylus (a pen with a radio frequency component) which allows the Tablet PC to detect where the stylus is positioned, as long as it is in some close proximity to the screen. As it hovers over the Tablet PC, the stylus can resemble the

¹InPlay technologies has published an excellent white paper on their web site (http://www.inplaytechnologies.com/tech_pen.php) explaining the differences between the two types of digitizers.

²We will introduce the ink stroke in chapter 5 when we talk about the type of data we will collect to help characterize Tablet PC usage. Microsoft defines an ink stroke as a set of properties and point data that the digitizer captures that represent the coordinates and properties of a known ink mark. It is the set of data that is captured in a single pen down-move-up sequence. <http://msdn.microsoft.com/en-us/library/microsoft.ink.stroke.aspx>

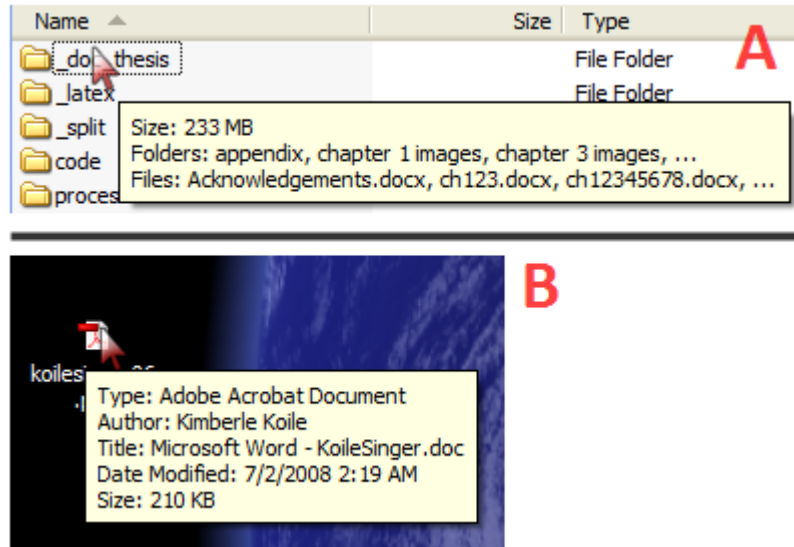


Figure 3.1: Hovering with the mouse over objects on the desktop provide different bits of information. A) mouse hovered over a folder of files B) mouse hovered over a PDF document on the desktop

mouse hover function. This is becoming more and more useful as information is displayed by pointing at objects on the desktop (figure 3.1).

The passive display is touch-sensitive and by using a stylus or finger, the user can navigate through applications. Even with a stylus, the passive digitizer will still detect any pressure placed on the screen, such as from a wrist when in a normal writing position. In this way, the biggest challenge to novice Tablet PC users is to allow their hands to rest upon the device without concern that the digitizer will pick up unintentional input (figure 3.2). When the pen stylus is turned upside down it can provide functionalities such as erasing (figure 3.10). This is not the case for the passive displays. Fujitsu and a number of other companies selling Ultra Mobile Pocket Computers (UMPC) have integrated a palm rejection [65] passive display. While it improves the writing ability with the pen stylus, the overall performance and sampling rate is at a much lower quality than the active digitizers. There are also Tablet PC devices that have recently been put on the market that offer both types of digitizers.

3.3.2 Slate, convertible, and hybrid

There are three types of Tablet PCs available on the market (figure 3.3). A slate refers to a Tablet PC that resembles a writing slate which loosely defined is a piece of flat material used as a medium for writing. Slate Tablets do not have a keyboard built into the unit and are primarily built for portability, both in size and weight. The convertibles or clamshells are the tablets that include a keyboard, a rotating display, and a possible optical drive. The convertible tablets are heavier and bulkier than a slate; the display swivels 180

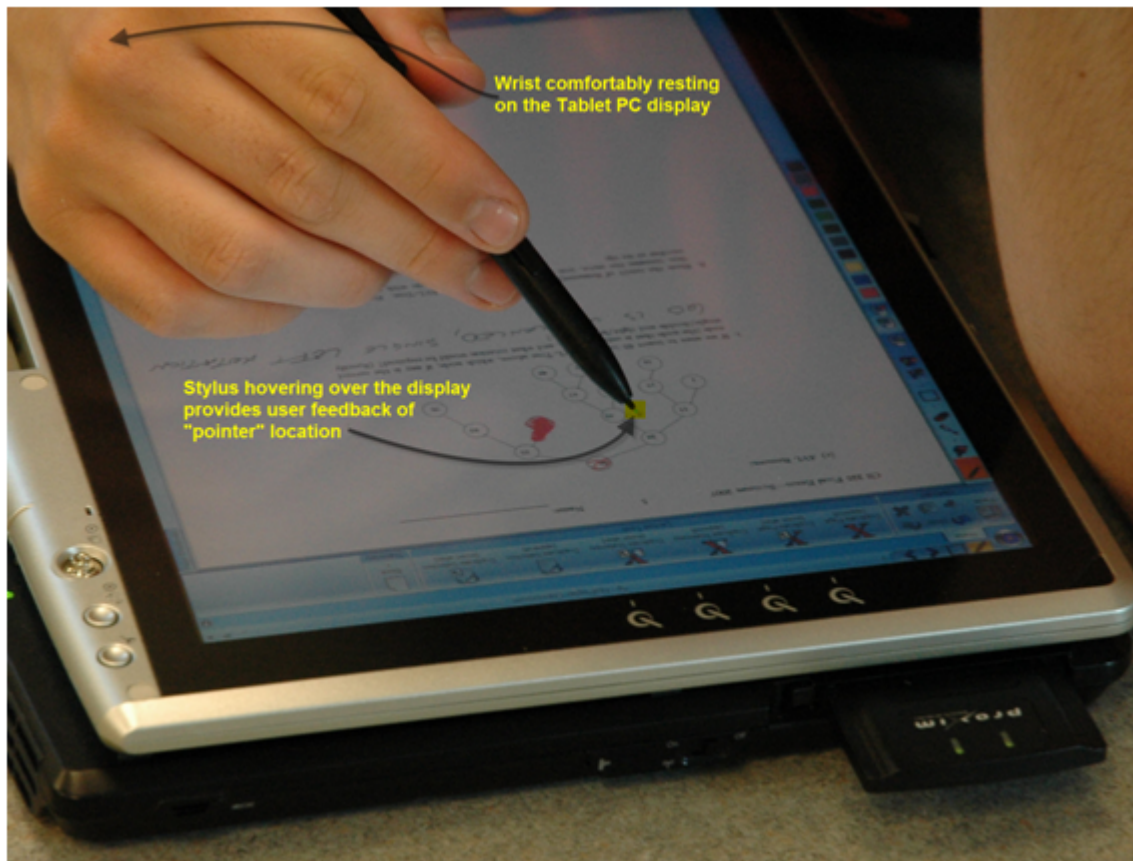


Figure 3.2: Example of student taking an exam. The pen is currently hovering over the display and the student's wrist is comfortably placed on top of the digitizer (LCD screen).



Figure 3.3: Three types of Tablet PCs.

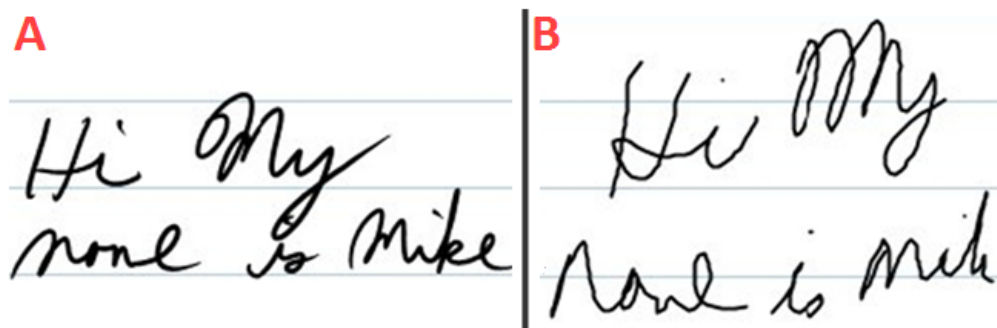


Figure 3.4: A) sample drawn on a active digitizer B) sample drawn on a passive (resistive) digitizer (source http://jkontherun.blogspot.com/jkontherun/2006/11/frequent_commen.html)

degrees to hide the keyboard, creating a thick slate that provides the addition of the keyboard and track mouse but not the feeling that one is writing on a piece of paper. The hybrid is a slate with a detachable keyboard. (It seems that the hybrid Tablet PC is a dying breed. Hewlett Packard has discontinued the TC1100 model, which was the only hybrid on the market.)

3.3.3 Rich ink and digital ink

Rich ink refers to Microsoft Rich Inking technology associated with the Tablet PC operating system. Many people still use the term “digital ink” to refer to ink strokes created on a computer, however that term does not distinguish between the inking technology of the past and present. The keyword “rich” refers to the vast amounts of information collectible by the active RF digitizer. The result is a smooth feel and appearance of ink (figure 3.4).

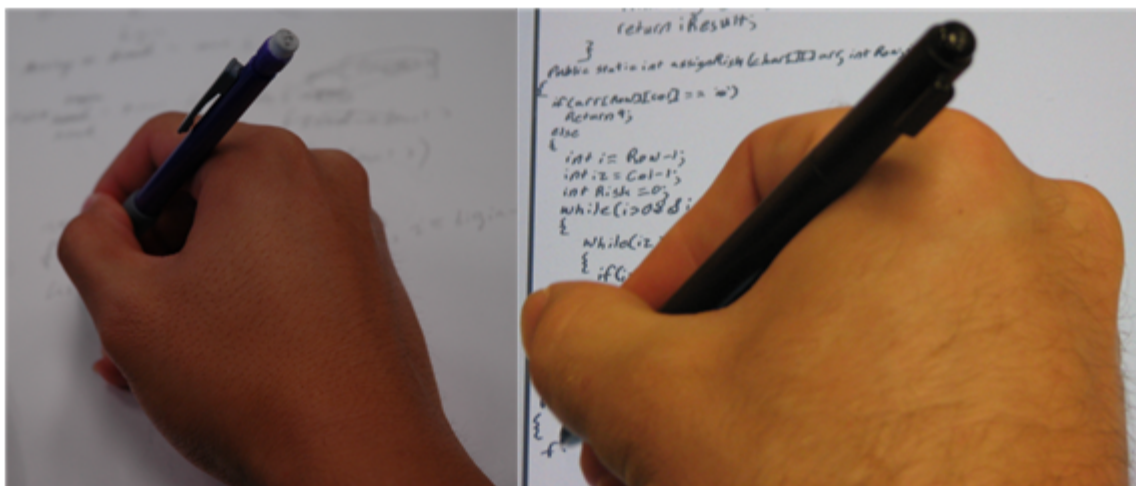


Figure 3.5: Comparison of two students writing. On the left the traditional pencil and paper and on the right, a Tablet PC.

3.3.4 The pen and paper modality

The active digitizer greatly improves the readability and ease of writing with a pen on a computer screen (figure 3.5). While the slate and hybrid Tablet PCs provide more of a paper-like “feel” than the larger and bulkier convertible Tablet PCs, the performance of the digitizer is the same in all models. See Lidström’s thesis [116] for an excellent comparative study between two input modalities, the mouse and the pen stylus. Liao et al. [114] investigate how to combine the advantages of physical artifacts like paper with the convenience of an electronic communication and archiving infrastructure such as UW Classroom Presenter (section 2.8.1). The paper addresses the differences between the digital canvas representing paper, and the inherent characteristics of physical paper. One example where the flexibility of the electronic expansive canvas can become a nuisance is illustrated in figures 3.6 and 3.7. During the spring 2007 semester (chapter 6) the user interface on the whiteboard application provided a zooming feature which provided an unintended mechanism to increase the writing space on the CS125 final examination. “Real” paper does not expand dynamically (like it probably could in the movie “The Matrix”). The problem with allowing a whiteboard page to dynamically grow (i.e. to be extended *well* beyond the default 8.5 x 11 paper size) is when one attempts to print the document. Either the page is scaled by “shrinking the image to fit in a printable area” and this might render a page unreadable, or the document needs to be reflowed [47]. The reflowing of ink strokes is not trivial and can require a sophisticated transformation of the student whiteboard document in order to render a coherent and readable PDF document.

3.4 Common Myths

3.4.1 Ink-to-text conversion

At the top of a Google search on the phrase “Slate Tablet PCs” was a link to <http://mobileoffice.about.com> with the following quote: “Slate Tablet PCs provide a great way to use handwriting recognition software and cut down on using a mouse and keyboard.” One of the biggest misconceptions about the Tablet PC is that the pen input is great for converting your ink strokes to ASCII text. No. If you want to write a term paper, and you usually use the keyboard as your primary input device, then use the keyboard! The pen input on the Tablet PC is the closest match to the traditional pen and paper modality. As such, it is “ink for ink” that makes the Tablet PC so useful. Just because your ink strokes are not displayed in textual form does not necessarily mean they are not searchable. Handwriting recognition is not a reason to purchase a Tablet PC.

Activities that are normally performed with a pen or pencil are good candidates for a Tablet PC. Technologies that do not require too big a behavioral change are the technologies that have the most promise to be integrated into the classroom. When Microsoft unveiled the Tablet PC, members of the development team were heard repeating phrases such as:

- “A tablet makes ink rock”
- “Ink is the focus, not handwriting recognition”
- “Ink is a first-class citizen”
- “Ink as ink”

An interesting facet to using the Tablet PC as a substitute for pencil and paper is that you can record the creation process. When a student hands in an exam written on the traditional pen and paper, what you see is what you get. You might be able to decipher remnants of erasures and crossings out of material, but not much more. You do not know how many times the student erased an answer only to rewrite the same response, three times, at the five and fifteen minute point, and just after time was called. The paper exam does not allow you to calculate how much time a student spent on a given problem or how many times they navigated back and forth between pages. Did they navigate between all the pages in the exam before handing in their exam to check their work? It is a lot easier to lose a paper exam and a lot harder to cheat on an electronic one. As a case in point, in one of our pilot studies, on two occasions, a student tried to convince the instructor that they had indeed completed a page of the exam contrary to what the archived printout of the exam presented. A quick analysis of the log file revealed that the student navigated to that particular page three times during the exam, and had not written a single ink stroke all three times.



Figure 3.8: Fujitsu Slate Tablet PC in a docking station. The Department of Computer Science at the University of Illinois at Urbana-Champaign invested over \$100k to purchase 30 plus earlier versions of this slate tablet.

3.4.2 Tablets are not as powerful as a “real” computer

Tablet PCs are powerful laptops. A Lenovo ThinkPad X61 Tablet PC convertible notebook with a DVD/RW (CD/DVD optical drive reader and writer), with Intel Core 2 Duo 1.6 GHz Processor, 4GB of RAM, and 120GB hard drive is quite powerful and costs about \$1500.00 (without an educational discount). As Tablet PCs become more pervasive, their size and price will continue to come down. It is also important to note that Tablet PCs almost always have a docking station (figure 3.8) which may contain an optical CD/DVD/RW drive (for those smaller Tablet PCs that do not have the optical drive built in) and provide a flexible stand which elevates the screen of the Tablet PC acting as an LCD display. Attach a keyboard and mouse and you have a powerful workstation. A number of universities have mandated the purchase of Tablet PCs for incoming students. They must believe that the devices are powerful enough for a four year education.

3.4.3 Difficult to use

As Tablet PCs become more pervasive, research on pen-input gestures and interfaces continues, and the Windows Vista Operating System (which comes with built-in support for Wacom enabled digitizers) improves, the user interface and applications will become more Tablet PC friendly (pen-centric). Applications written and designed for the Tablet PC will become easier to use. However, the myth “difficult to use”

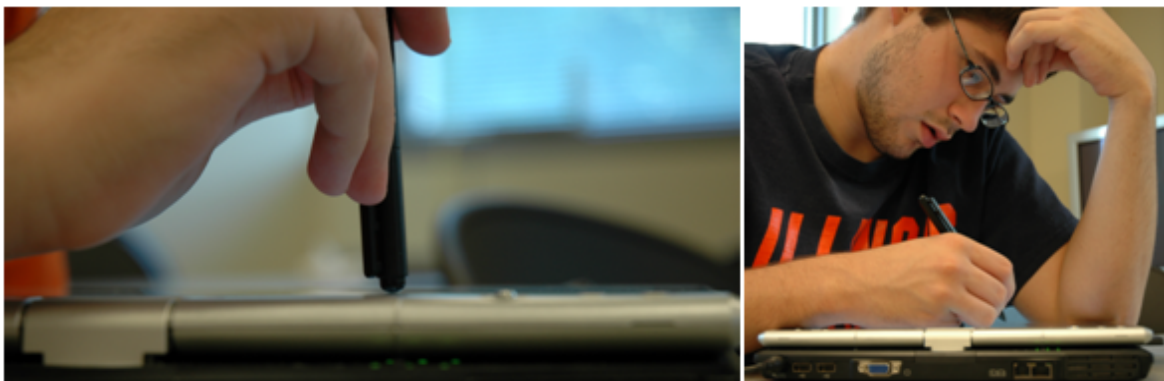


Figure 3.9: Examples of in-class use of the Tablet PC during instruction in one of our pilot studies. The focus is on completing the exercises, not the technology.

refers to the fact that many instructors, for example, are afraid to integrate this new technology into their classrooms for fear of technical failure. If you can draw on a chalkboard then you can write on a whiteboard with markers. If you can do the latter, you can certainly use a pen-stylus (figure 3.9). Not only are the digitizer pens more natural to write with than chalk and chunky whiteboard markers which leave unwanted residue on the hands, but the erasure for the Tablet PC is quicker, easier to use, and more sanitary. In a nutshell, if you can write with a pen and paper, you can do the same on a Tablet PC. On one occasion the Department of Computer Science received a request from a student wanting to observe a Computer Science class as part of an assignment in one of his courses (he was not a Computer Science student). About 10 minutes into the lecture this student appeared at the entrance to our laboratory to “passively” observe the lecture. I motioned to the student to take an empty seat. With the help of one of the students sitting nearby the vacant seat our visitor (unbeknownst to me at the time) used the Tablet PC to articulate his observation. An excerpt of the student’s observation is illustrated in figure 3.10. (The entire sample is included in Appendix ??). Koile and the CLP group provides a glance of first graders using Tablet PCs in class [107].

3.5 Exploration of Note-taking on the Tablet PC

The sub sections below provide a discussion of a number of features available through the different modes of note-taking in a class session.

CS Tablet PC... surprisingly easy to write on this thing... fairly accurate. When I first came in, I caught sight of the teacher... I believe his name is Chad ~~Piper~~ Piper. He is young, compared to my MCB teachers... probably about 30 yrs. old. He motioned me one of the 3 open seats ~~to grab a seat~~. About one person looked familiar to me. I'd seen him

Figure 3.10: Example of a student observing a class session during our spring 2007 pilot study (chapter 6). This student was not a student in Computer Science and to our knowledge had never seen a Tablet PC before entering our laboratory classroom.

3.5.1 Writing from scratch—starting from a blank “slate”

One measurement of student interaction in the classroom would be as simple as detecting if, when, and how a student interacts with their tablet. Is the student currently within the window of the note-taking application or have they potentially disengaged to play a game or browse the web? If they are browsing the web, what is the URL and, is there a chance the URL is related to the current topic? At what point in the lecture did the student leave the scope of the note-taking application?

One advantage of the Tablet PC over the traditional pen and paper in the classroom is the ability for an instructor to electronically monitor what and when a student is or is not doing something. We use the term “something” because we cannot be certain that interaction with the Tablet PC means a student is paying attention or engaged in the lesson. However, if the instructor wanted to know how many students were taking notes at any given point in the lecture, they could do so by querying the number of students running the note-taking application (which happens to also automate the taking of attendance). We implemented this functionality in our spring 2007 study (chapter 6). Without tablets, an instructor would be forced to come up with a rough estimate after a cursory glance of the students in the class with pen and paper or some other electronic device. An important question to consider is how does an instructor (in the traditional setting) gauge the level of interaction and or interest of the students?

Research has been completed attempting to detect a student’s motivational state during instructional interaction [49–51] to benefit the performance of Intelligent Tutoring System (ITS) [5, 170]. In a traditional instructional setting or any other social interaction there is a vast amount of information available through various communication channels, such as facial cues, posture, eye contact, head gestures, etc. The authors make the point that many such cues that help us detect other people’s emotions (or motivation) are perceived

unconsciously.

Using tablets, we can automate the capture and recording of information for later analysis, review, and reporting. Accumulating this information manually takes more time, is inaccurate, and forces the instructor to make mental notes so that they can record (if they so wish) information after the lecture has ended. The disadvantage of pen and paper is that we cannot seamlessly query the class, and from a distance, we certainly cannot monitor a student's ink strokes:

- How many pages of notes has the student produced for a given class session?
- At any given time during a lecture how many students are writing? At what point in the lecture did the traffic of writing on the Tablet PC peak?
- Can we make associations between the characteristics of a user's inking and their comprehension, interest, or engagement with an objective in the lesson?

Figure 3.11 is an excerpt of a high school student's Algebra 2 exam. This student had spent a great deal of time and ink (we show how this measurement was determined and presented in chapter 7) on this page of their examination. As the instructor caught notice of this statistic (a clear outlier in the class) he navigated to the student's whiteboard to see what he had written. The instructor was slightly amused, but when bringing this to my attention (he wanted to keep a copy of the drawing since the student would inevitably erase the doodle) I reassured him we had an archive of the activity in the logs and he assured me that this student doodles in the process of learning and discovery. I was amused.

3.5.2 Slide synchronization of instructor white-board contents

The association of material from an instructor-slide with that of a student's annotations (or just general tablet interaction) can provide a finer granularity of user interaction based on specific objectives taught within the lecture. This assumes that an instructor has defined and specified a list of objectives to the system such as those illustrated in figure 3.12 (we provide further explanation of objectives in section 6.2 of chapter 6).

Which slides does the student interact with; more specifically, which slides does a student add annotations to? Does the student move ahead or backwards to review material at a different pace than the instructor? What region of the slide is the student annotating atop? In the previous paradigm (the blank white board), where (in terms of X and Y coordinates) the student chooses to annotate is a matter of personal note-taking preference; however, if the slide synchronization interface provides a suitable means of interacting with the instructor slide content, we can infer a great deal of information.

Two different interfaces for slide synchronization provide various means of attaining interaction-type information. The first is an interface where there exists a single whiteboard window for taking notes. This

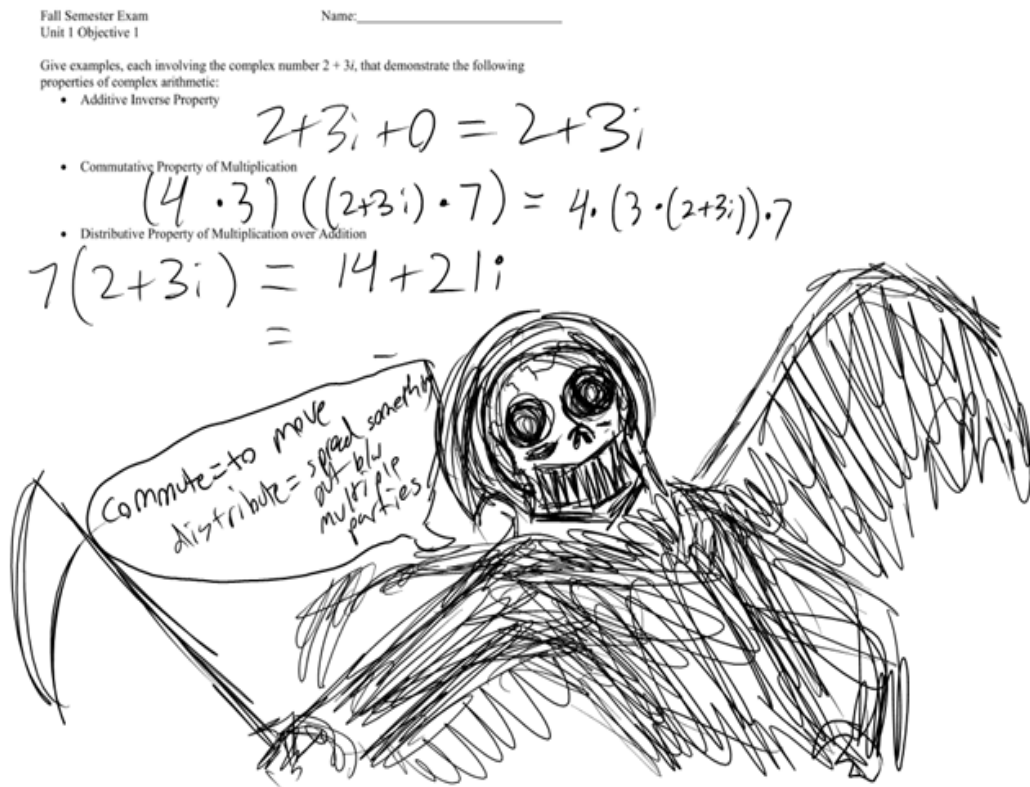


Figure 3.11: This high school algebra 2 student had an abundance of ink (we illustrate how this measurement was determined and presented in chapter 7) on this page of their examination. Upon taking notice the instructor's curiosity moved him to navigate to this student's whiteboard. He was slightly amused, but reassured me that this student doodles in the process of learning and discovery.

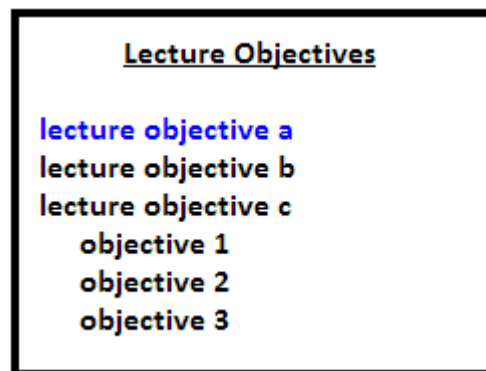


Figure 3.12: Example of lesson objectives

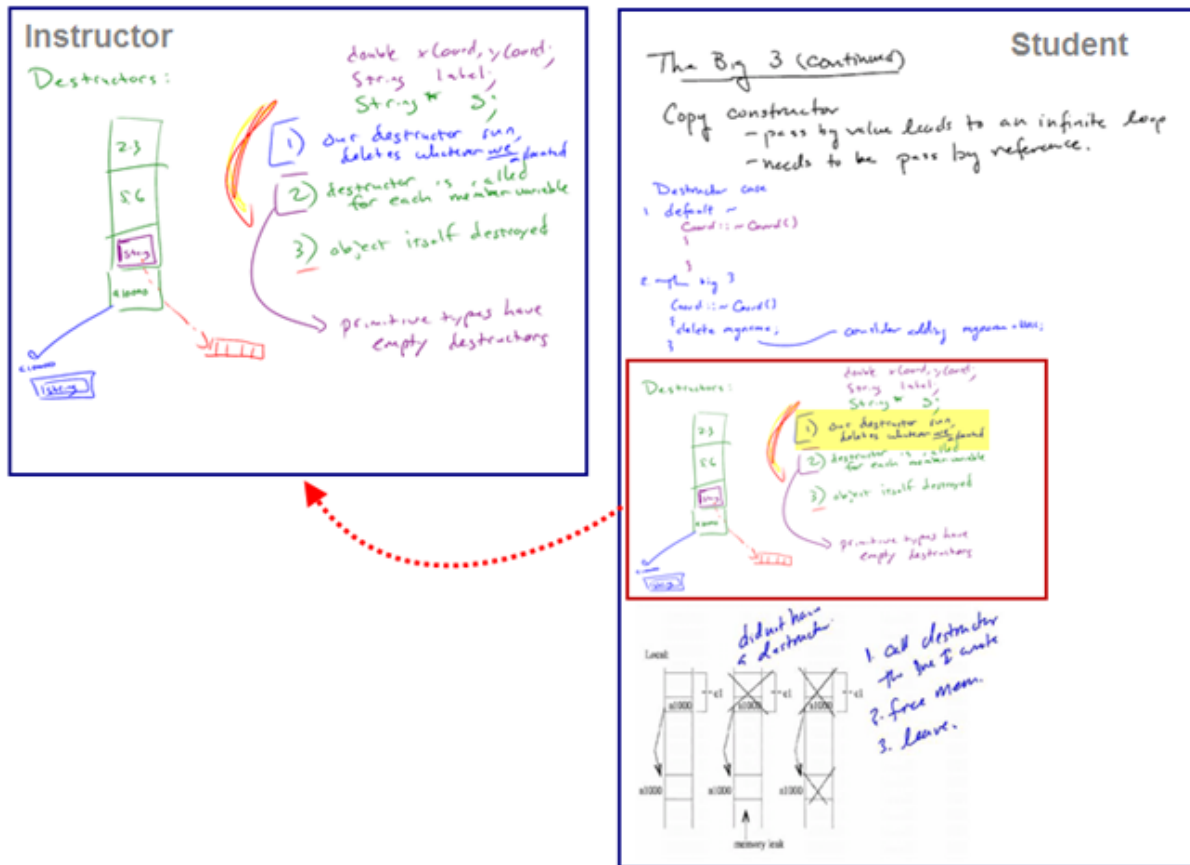


Figure 3.13: Instructor slide content copied to student private workspace. In this interface the student does not have to worry about an instructor's whiteboard annotations intersecting with their own.

mode of interacting with the whiteboard is similar to the interface we implemented in the Beethoven demo at the Siebel Center Grand Opening [143] (see section 3.7). A second interface uses two panels; one panel is a read-only scaled version of the instructor's whiteboard slide (on the left-hand side of figure 3.13) and the second panel (the oblong panel on the right) is the student's private whiteboard for taking notes. This is similar to the interface we implemented in the first version of e-Fuzion in 2000 where the students were forced to interact with the instructors panel in order to copy instructor contents to their own space (see chapter 4 section 4.2.5).

In this example, the copying of instructor material is accomplished by selecting objects in the instructor read-only panel and dragging them to the personal notes space, or by pressing a button on the instructor panel which automatically copies and pastes the content into a position in the personal note space. Recall that the timing information for all tool usage is recorded during the lecture. As a result, contextual information is available for annotations created by the student. For example we can compare what a student was doing before, during, and after actions performed by the instructor and other students.

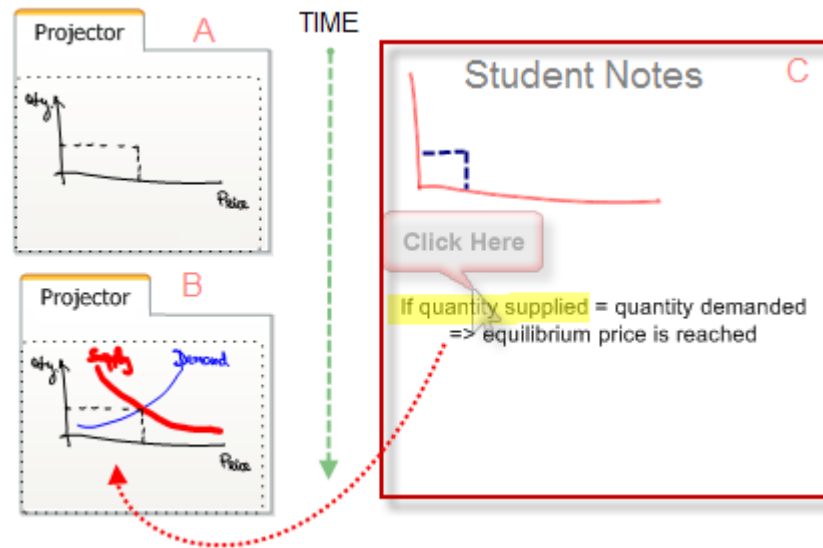


Figure 3.14: A) Display of instructor's whiteboard at time x. B) Display of instructor's whiteboard at time y. C) A student clicks on a highlighted ink stroke written at time y to display what the instructor's whiteboard at the same time.

By making implicit associations between instructor and student interactions, we can enhance the experience of reviewing notes outside of class. For example, if a student wanted to remember why she highlighted some of the instructor whiteboard contents copied to her notes area, she could ascertain what was being projected on the screen, since clicking an ink stroke in the student's notes space would also generate a snapshot of the instructor read-only whiteboard projected at that time stamp (figure 3.14). This could be an important feature when reviewing personal notes post lecture, and is certainly not possible in the traditional classroom without Tablet PCs. As mentioned previously, eClass [88] generated a time line of instructor activities, but it was never associated with student interaction (because their system recorded instructor presentations and only student audio responses) and so the indices created by the system were impersonal and only used to navigate to a point in the video indexed on the instructor's annotation.

The ability to correlate student-activity with instructor-activity at the granularity of an instructor's ink strokes or tool usage in the application is a novel tool. It also provides a solution to the recommendation from eClass that "...the integration needs to be smarter...by providing a more semantic linking of the notes."

If an instructor moves at a pace not conducive to a student's level of interest (too fast or too slow), a student might navigate to previous or future slides (if available to the student before the instructor first presents them) to review old or skip ahead to new lecture material. How much time a student spends on a particular slide can provide additional information that was not available without slide synchronization. Visualization of the aggregation of such information could be insightful for instructors during and/or after the lecture (see chapter 7). The addition of the instructor's lecture material can also help us to measure

a student’s level of interaction of instructor-created material. In a traditional class setting, instructors are limited to human-type cues such as facial expressions, body language, and obvious interactions—whether with technology, pen and paper, or a person sitting next to them—to infer the motivational state of the student during the interaction.

Previously, we assumed an implied association of student ink strokes with an instructor’s current objective. So, if a student navigated to a previous slide to add additional notations (which includes erasures), we might incorrectly assume that the added content is related to the current objective or the previous objective when the annotation was originally created. Regardless of the rationale for making such a change, the system can record that such a change occurred at time X where the instructor was talking about topic Y. Similar to the concept of the eClass project, a context is created for the in-class lecture for later review by students.

How a student interacts with the content of an instructor’s whiteboard provides additional contextual information that could not be gleaned from a user’s private whiteboard, and certainly not from a piece of paper. That is to say, in the previous note-taking modality, we could determine if a student was writing or not; however, with slide synchronization, the current slide with which the student is “interacting” provides additional information not available before.

The ability for a student to create questions from portions of instructor’s slides can provide insight into a student’s doubt. Just as annotating on a particular slide provided insight into what specific material a student was interacting with, so to, the selection of material from either an instructor’s or student’s slide(s) (in collaborative learning environments) to formulate questions provides indication that a purposeful interaction with lecture material has occurred. We present an example of this functionality in section 4.2.5.

Another example of whiteboard distribution is an approach where the instructor can decide on-the-fly what contents to send to students (this includes the possibility of not sending ink to the room display). As an example, instead of muting the room display so an instructor can solve a question as the students are also working on the same problem, the instructor can keep the contents of the screen on the projector display (i.e. the assessment itself), without displaying the answer as it is being generated by the instructor in real-time.

3.5.3 Association of interactions

The data recorded on each page of a student’s whiteboard documents provides a number of novel possibilities. One is the ability to “TiVO the learning process” allowing for the replay of all interactions recorded on the Tablet PC whiteboard application. In a networked class environment, we would have the ability to correlate ink-data created by all users in the class to provide a meaningful context. For example, if an instructor had

provided a set of objectives for the lesson, and each page of the instructor’s document was associated with an objective, student annotations could automatically be associated with this type of meta-data produced by the instructor’s Tablet PC. This provides a more meaningful record of interaction than the raw time data provided by the ink stroke information.

Another possibility is region-based detection of ink activity (an indication of some level of engagement) in certain X and Y coordinates on a page. As an example, consider a work sheet laden with areas where the user is required to write in an answer (answer boxes) to an assessment. DyKnow® (section 2.8.3) and Classroom Learning Partner (section 2.8.1) have implemented some of this functionality. The absence of ink (such as a measure of idle time), measurement of ink, how quickly the ink stroke was created, and when the ink was inserted, provides a history of user interaction in a specified region. The time a user spends on a page of a worksheet (like an exam question) with/without writing might provide insight into a student’s ability to answer a question/exercise. Without regions it would be more difficult (but not impossible) to differentiate between what portion of the page a student worked on for some duration.

The Tablet PC allows us to more easily record different types of interactions that would be more difficult with traditional pen-and-paper. For example, we might want to distinguish between the amount of time a user spent writing and erasing. By calculating the “idle time” (a measurement of the time a page is displayed without interaction by the user) we can differentiate between time spent writing and time spent **not writing**. It might be the case that a student is staring off into space while on a given page of a document. In this scenario, the spectator time is not indicative that the student is thinking or working through a problem in their head. The same problem exists in the paper world.

3.6 Instructor Role

In this section we present a number of examples how the Tablet PC can facilitate learning even when used as a presentation device where students in the class do not have computing devices. We expand on these examples in chapter 5.

3.6.1 Example of whiteboard distribution

At the grand opening of the Siebel Center for Computer Science (the implementation is discussed in greater detail in section 3.7) our research group called SLICE (Students Learn in Collaborative Environments) taught 20 minute class sessions demonstrating our view of the “classroom of the future.” Just before the start of each class session all of the Tablet PC and PDA devices in the room were pre-loaded with the

lecture presentation and connected to the e-Fuzion server. The connection to the server provided for slide synchronization and the transmission of instructor annotations. Annotations written by the instructor were pushed to all of the student machines. As an example, the three Tablet PC displays in figure 3.15 contain the same content as that projected on the screen at the front of the classroom. When an instructor changed a

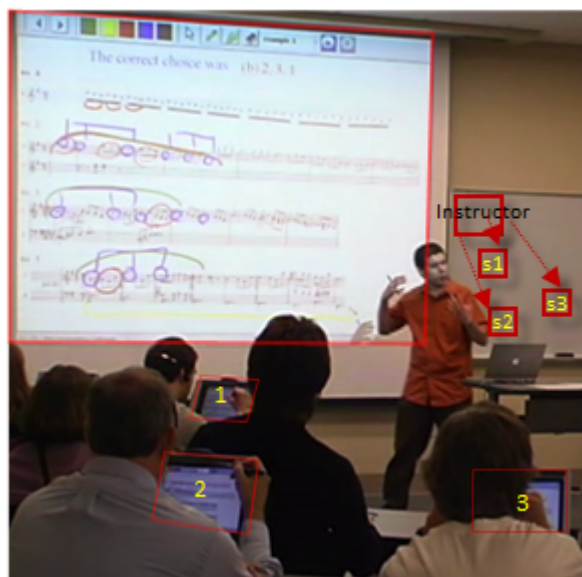


Figure 3.15: Siebel Center grand opening classroom of the future lecture demonstration

page (unless a student was in the act of writing) the page displayed on the student’s machine also changed. In this example of slide distribution, if a student came into the class and *never* “touched” their Tablet PC, their display would look identical to the projected display of the instructor. Consider the two images displayed in figure 3.16. The difference between the image on the left and right is that the instructor navigated to the “next” slide and as a result, changed their “current” page. This change was reflected on the classroom display (since the instructor’s tablet was connected to the classroom projector) and all students’ displays. This is one illustration of slide synchronization. The only exception, however, was if a student was in the act of “writing” (annotating their own whiteboard documents) while the instructor changed pages. In this scenario, the change of page occurred after a student completed their annotation (as soon as the student’s pen stylus was moved away from the digitizer).

The user interface implemented for our classroom of the future demonstration was simplified to only allow page navigation, drawing and erasing of ink strokes, use of a highlighter, and a play and stop button for listening to musical excerpts. Students were not given the ability to insert or delete slides in their documents. This is just one example of sharing instructor slides with a class of students; the instructor has a static document to teach from and when an instructor writes on a page, whatever shows up on the projector

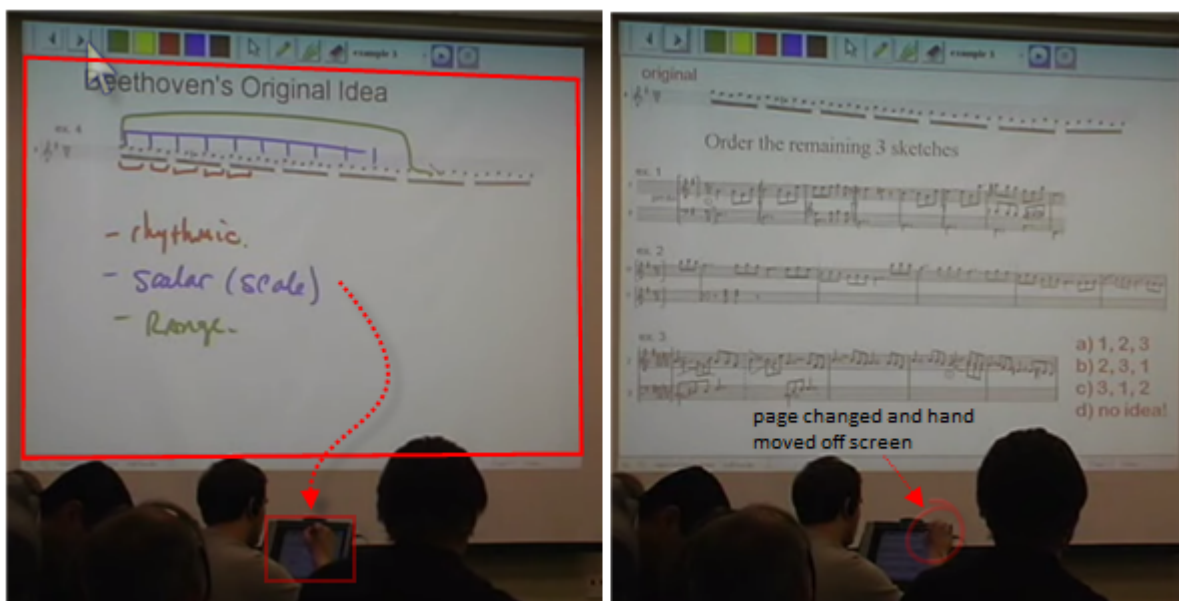


Figure 3.16: Example of slide synchronization

(the instructor's whiteboard) is sent to the student's whiteboard. However, students could write on their devices without the contents being sent to the instructor (figure 3.17).

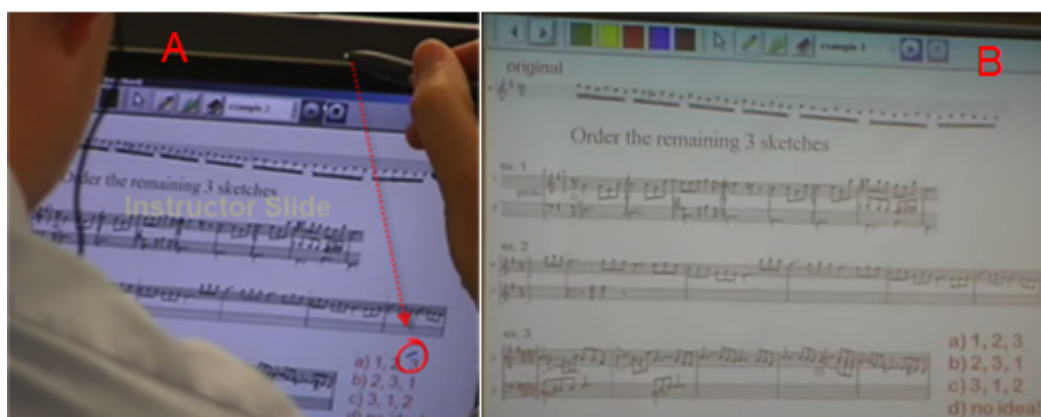


Figure 3.17: A) Since the client machines all received the original instructor presentation, only instructor annotations were transmitted during the class session. The annotation inscribed in the lower right hand corner was made by the student. B) The instructor's version of the slide contained no annotations and so none were transmitted to the student's display on the left.

One disadvantage of this mode of slide distribution is the possible intersection of annotations that can occur unintentionally. Imagine that the “pre-loaded presentation” for our demonstration lecture is the one-page document shown in figure 3.18. First we pretend that the sending of instructor annotations is temporarily disabled. Furthermore, let's assume that the image on the left-hand side of figure 3.19 represents the instructor's document (whiteboard state) while the image on the right represents a student's document.

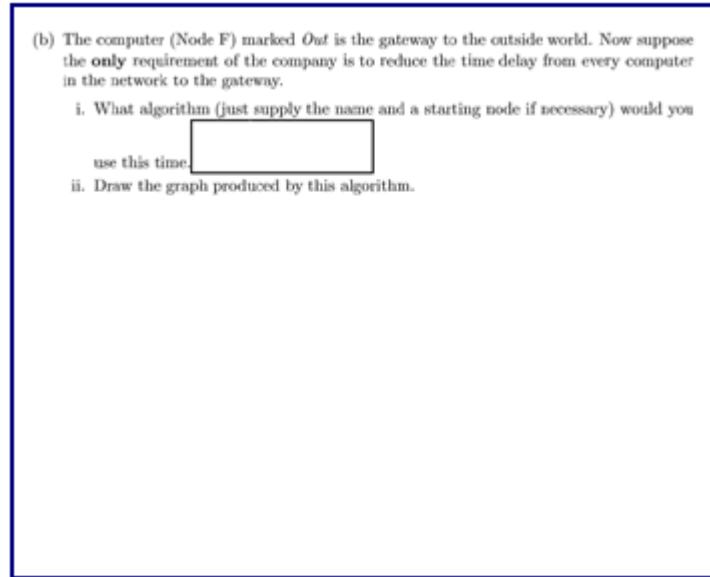


Figure 3.18: Original “pre-loaded” whiteboard document

If we suddenly enabled the sending of the instructor’s annotations then figure 3.20 illustrates a potential state in the student’s whiteboard. Although sending of the instructor annotations would have caused collisions and hindered readability *while* writing on their whiteboard, this example illustrates the potential problem of this mode of slide distribution. Although the text of the background is readable (the two whiteboards contain the same text in the same location so the textual blocks are literally layered one on top of the other giving the illusion of one textual passage) the differences in ink location caused “clashes.” When the instructor annotations were pushed to the student’s document, the layering of the two whiteboards causes a number of “collisions” to occur on their respective displays. This results because the contents of each whiteboard were created in isolation of one another (i.e. we did not send instructor annotations to the student whiteboard). However, even if the two students were inking in real-time on a single whiteboard space (as is typical in whiteboard collaborative systems [81, 167, 169]) there is no guarantee that collisions would not result. Network latency can cause a delay in the synchronization of ink strokes between the two devices.

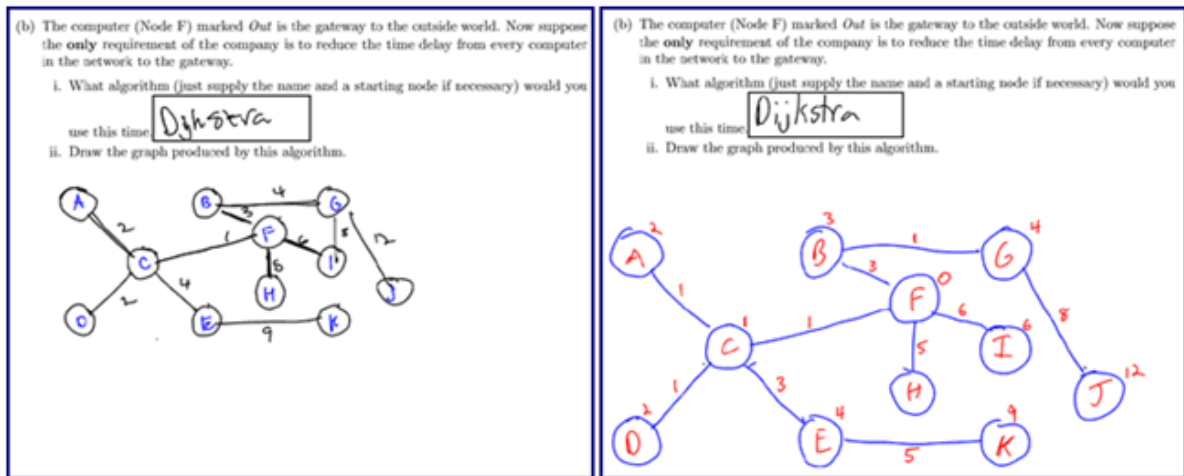


Figure 3.19: The state of two whiteboards without sending instructor-created annotations

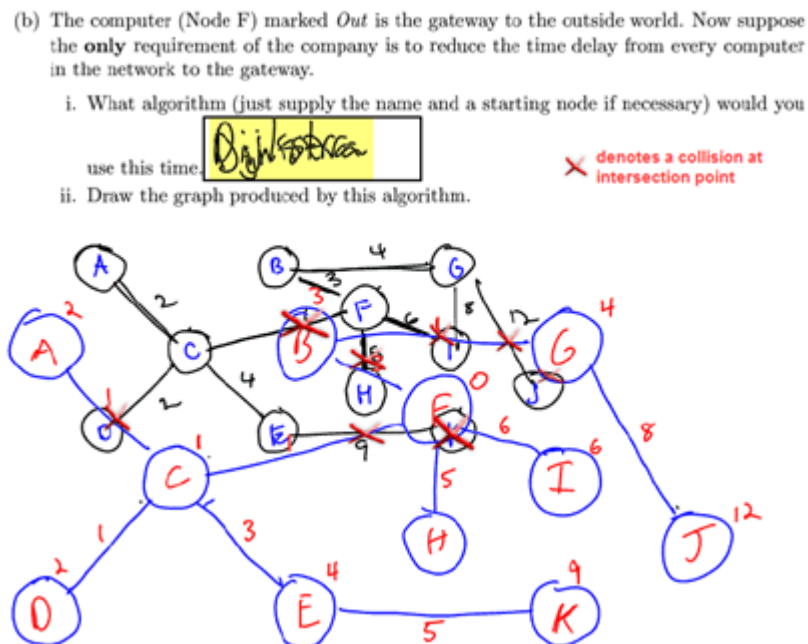


Figure 3.20: Example of “ink collisions” when sending instructor annotations to student whiteboards

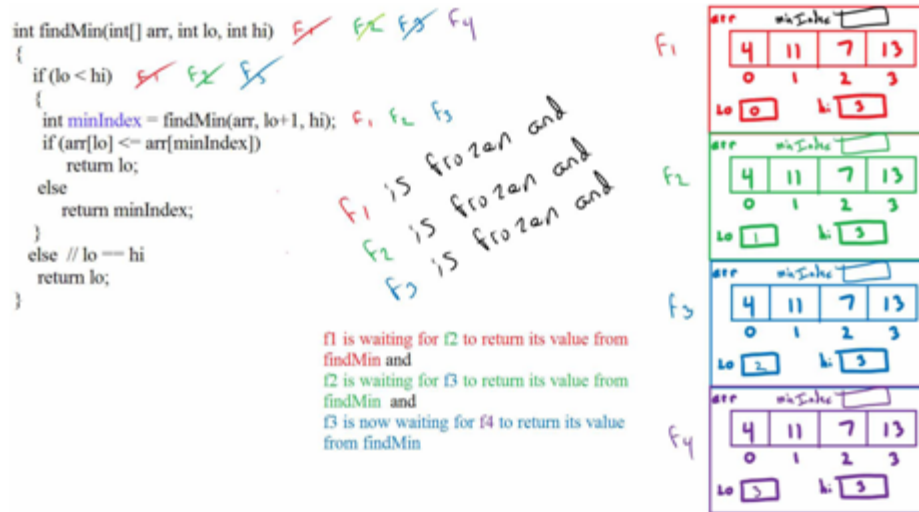


Figure 3.21: An example of color emphasis of subsequent method calls

3.6.2 Points of emphasis

Points of emphasis refers to features in the whiteboard application that provide an instructor with tools for visually enhancing their lecture content. Such tools include the laser pointer³, highlighter, and annotations such as circles or check points that can be associated with whiteboard contents—whether they are bullet points, previously drawn ink strokes, or inserted images. Color can also be used as a means of placing special emphasis without marking up a page using annotations. Consider an excerpt from a CS125 (Introduction to Computer Science) document created by the instructor in the fall 2006 semester (figure 3.21). This document was actually created outside of lecture (see appendix for complete document) and took close to two hours to complete. This lecture example included the tracing of a recursive method: distinct calls to the method `findMin(int[], int, int)` are distinguished by color. A refined user interface would have to be developed to allow an instructor to seamlessly (implicitly) create such an example on the fly. Teaching is not a straight forward process. While an instructor can have a clear plan of action (such as a list of objectives and examples to present) student questions can easily derail the lecture. It is then up to the instructor to get back “on track” so that he or she can complete the lesson’s objectives. In a hypothetical class where there are no student questions the instructor would still have to plan ahead of time how to dissect and organize the lecture slides. As an example, consider the situation where the instructor wants to demonstrate the difference between using the in-order-predecessor and in-order-successor in the remove algorithm for a binary search tree. Figure 3.22 was extracted from a document created by Jason Zych in the summer 2005

³To support the “electronic highlighting” of whiteboard contents we implemented a laser pointer tool. The tool worked by creating red ink strokes temporarily displayed to the screen. As soon as the pen-stylus was lifted away from the digitizer any ink strokes created with the laser tool were removed giving the illusion of the traditional laser pointer.

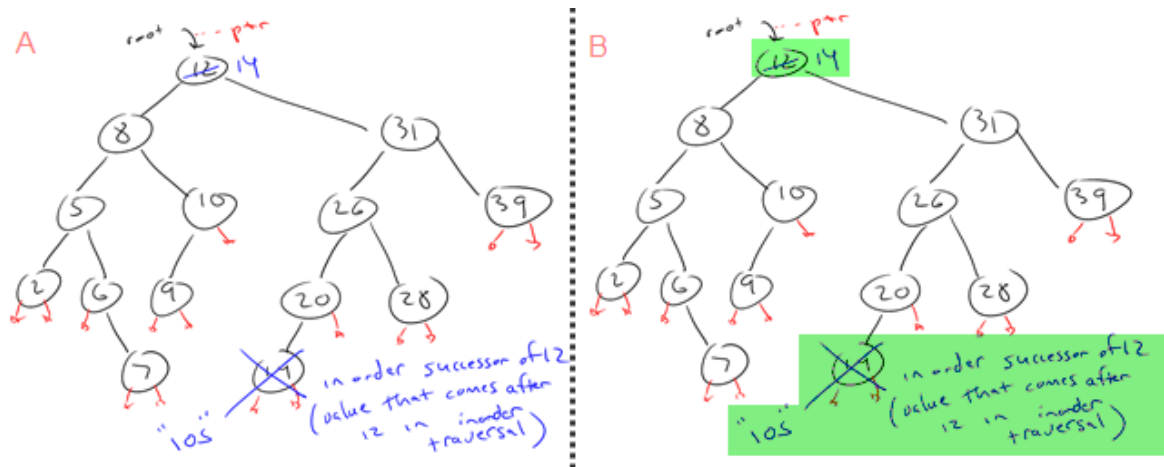


Figure 3.22: A) Binary Search Tree snapshot on page X of a document. B) After pressing the “duplicate page” button a copy of the page is inserted after the current page.

session of CS225. We traditionally required the students to use the in-order-successor when removing the root of a binary search tree. (The rationale was that it would be easier to grade 350 exams if all the students used the same replacement mechanism). After completing the IOS example let’s imagine that Jason used the “duplicate page tool” to set up the same example tree from the previous slide. Before continuing with the comparison, Jason would still need to edit the newly-inserted slide by removing the green highlighted content. The other option would be to traverse back some number of pages in search of the original tree. However, we still have the problem of how to insert the “original” slide in the “correct” location in the document.

3.6.3 Recording the teaching process

With a Tablet PC whiteboard application used in lectures, instructors have the ability to “record the teaching process” by taking snapshots of the whiteboard contents at key moments. The snapshots are created using a duplicate-page tool feature; at any point in time an instructor can create a copy of one state of the whiteboard (one page) and create a copy of the current page to be used as the starting point for the next stage of the process and next page of the document. This process continues by making a number of modifications to the state of the newly inserted duplicate page. When the instructor wants to preserve the differences between the previous slide and modified slide, she continues the duplication process. In this way, we can provide a post-lecture document representing the teaching process for students to review outside of class.

Among the suggestions presented by Abowd and Brotherton for future capture and access systems was the recommendation that “the quality of the captured materials needs to be of the same fidelity as presented [86].” Although they are referring in part to the quality of the audio and video recordings, they explicitly

state that the overall integration needs to be smarter by providing automated summaries of a lecture, or by providing a more semantic linking of the notes. Without providing additional semantic information in the form of meta-data (such as instructor objectives), correlating ink strokes on a single slide with audio and video lacks context for the reviewer. By adding objective-type attributes to ink strokes, such associations can automatically be generated. However, the instructor still needs a way to signal that he has moved on from one objective to another. As a result, it seems reasonable to assume that additional time and effort will be required by instructors in the preparation of and delivery of lectures. Smarter documents will be required for automating summary-based information and a greater amount of system interaction will be required to provide for a more exact record of a class session. However, these requirements are incompatible with the prerequisite that the system should be able to “...support generalized capture of lecture materials with no extra instructor effort[86].”

In sections 3.4.3 and 3.3.4 we tried to make the case that the Tablet PC is as natural and easy to use as a pen and paper. Imagine that we wanted to build an application that did not require additional interaction on the part of the instructor aside from the traditional whiteboard use. By doing so, we would have to compromise the quality of the archived class session.

As an example of additional interaction that might be explicitly required by an instructor, consider the following scenario: what if a student has a question that requires a subtle change in the original content of a page on the whiteboard? How does the instructor address the student’s question without erasing the old content first? Does she just insert a new (blank) page? The process and organization of the page information (as would be the case in the animation analogy) could be disrupted. That is to say, pages created on the spur of the moment might not be placed in the correct chronological location in the document. For example, an inexperienced instructor might append a page to the document to address a student’s question. If our concern is providing a post-lecture document that, to the best of its ability reflects what happened in lecture, the instructor will need to spend time editing and creating the document outside the class. This is far from automated. While it is possible to create all sorts of links to provide “bookmarks” to given pages, the expectation is some chronological order of pages that can both be navigated and printed, such as a PDF document, and we would like the document to “reflect what actually happened in the lecture.”

Consider instructors that choose to write from a blank whiteboard instead of importing or opening a previously created document. In this scenario, the instructor has the added responsibility of organizing his document so that it can be reviewed after the lecture by students. In the circumstance where a document previously created is opened, the instructor will still be required to organize the lecture notes (document) either during—when it is fresh in her mind—or after the lecture.



Figure 3.23: Tick tack toe board at the finish state

There is no easy way to automate the association of different objectives with a single instance of the current whiteboard. When the instructor uses the duplicate page tools, they are inherently associating a specific objective to its previous content (the contents of the page copied) and this is purposeful and not considered automatic. In the explicit act of duplicating a page the teaching process is interrupted. In an early version of our whiteboard application (e-Fuzion) in order for an instructor to duplicate a page they had to do so “manually” through the following steps:

1. Insert a blanks slide
2. Traverse to the source page
3. Select the contents to be copied
4. Navigate ”back” to the newly-inserted slide
5. Paste the contents in memory

For the sake of argument, let’s assume that an instructor is working on a single whiteboard page. When an instructor chooses to export her lecture as a PDF document for student review and printing, how do we automate the number of slides to be created in the animation paradigm? Let’s consider as an example a completed game of tick tack toe (figure 3.23).

At what point(s) should we create a “new” page? Should we create new pages for each subsequent ink stroke? Or, should we create new pages at some regular time interval. Tick Tack Toe is an easy example because the computer can differentiate between the different states and ink strokes; the initialization of the board contains two parallel vertical lines intersecting two parallel horizontal lines. Additionally, the ink recognizer can tell the difference between the letters ‘O’ and ‘X’. If the program only allowed for these symbols, the computer could implicitly dissect the whiteboard contents into pages of subsequently added ink strokes. (Anderson [8] has also looked at trying to decipher “stages” or “phases” of a diagram state, but not for the purpose of creating a static document for out-of-class review.)

Why would we want to do this? If you were a student reviewing the contents of the lecture and all you had was a single paged document (the final state of the whiteboard, in this case the tick tack toe board illustrated in figure 3.23), what could you decipher aside from the final state of the board and who won? Perhaps you have a keen memory and can recall the entire process from the beginning to the end. (Typically, lecture content is not as trivial as this example.) To understand the strategy of how ‘O’ won the game, it would be beneficial to have additional information available, such as which player (‘X’ or ‘O’) went first (despite the fact that ‘X’ usually goes first). From the above diagram, we can infer who went first since ‘O’ occupies the majority space on the board. But, what was the order of moves? What was X’s first move? The answers to these questions are the details hidden in the states of the diagram not archived in figure 3.23 and details that are available in the log files of our e-Fuzion application used by instructors for presentation (chapter 5).

So the question is how we can automate the creation of a static document from the final state of a single whiteboard page. We need to decide what granularity of data should be used to trigger the creation of a page frame. Should we assume that the granularity of the ink stroke is sufficient (every time the system detects another ink stroke it automatically creates a new page in our post-lecture document) or do we need to go even finer to a set number of seconds or even milliseconds? Figure 3.24 illustrates an example of how a time-based algorithm is not suitable since it creates too many extraneous slides. The highlight marks were added to emphasize the small differences of each subsequent state.

If the system created pages at the granularity of ink strokes (easy to be done) we would have a total of 18 states or 18 pages (figure 3.25). We assume that the letter ‘X’ takes two strokes, while lines and the letter ‘O’ take just one stroke.

3.7 Siebel Center Grand Opening

To demonstrate how we integrated e-Fuzion into our classrooms, we prepared a presentation for the grand opening of the Thomas M. Siebel Center for Computer Science at the University of Illinois. While we wanted to illustrate our vision of the classroom of the future using e-Fuzion, we also wanted to enhance our system to support pervasive computing including the integration of wireless mobile devices and remote classrooms (figure 3.26). To extend our learning environment we used bidirectional high definition video streamed through the ConferenceXP infrastructure [12] to several remote locations. We also refined the e-Fuzion architecture by creating a new ink distribution architecture. After the instructor had drawn an ink stroke, the system needed to perform a series of steps to prepare the ink stroke for transmission. Since

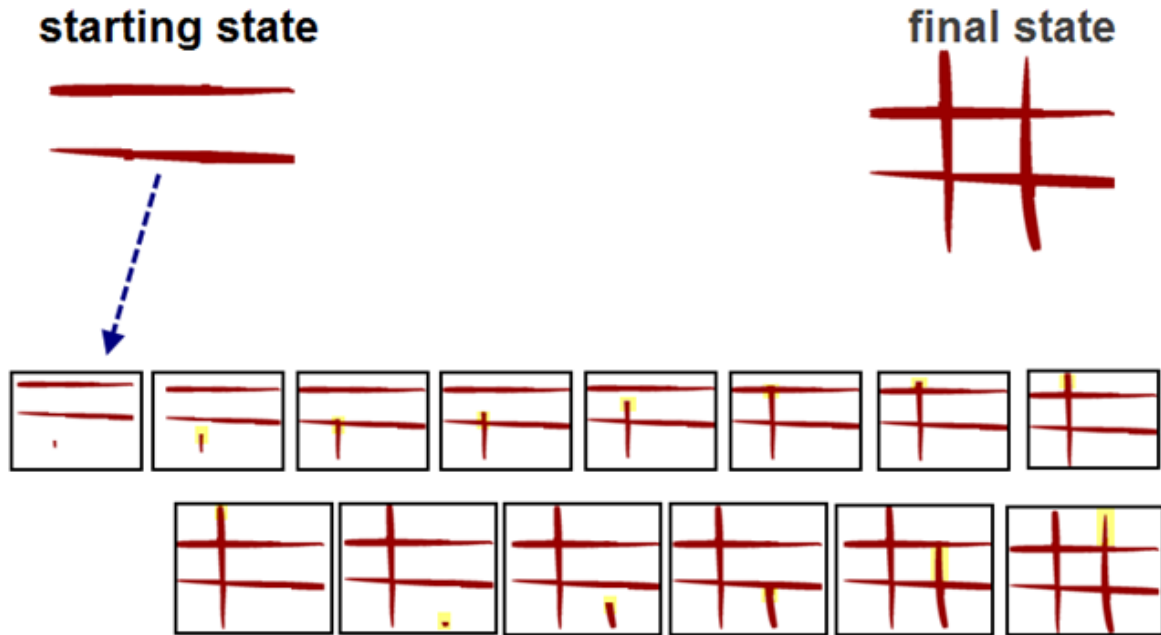


Figure 3.24: An example of how a time-based algorithm is not suitable to automate the creation of post-lecture slides. In this example it creates too many extraneous slides. The highlight marks emphasize the small differences of each subsequent state.

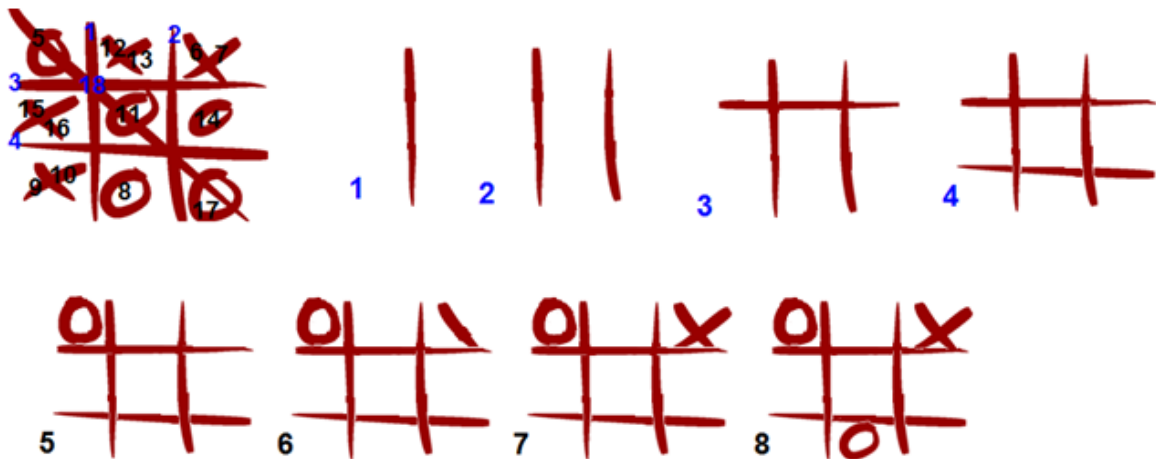


Figure 3.25: Assuming that slides are generated automatically by the detection of ink stroke creation, and that lines and the letter 'O' is just one stroke, and the letter 'X' is two strokes, we would have 18 pages created for our post lecture document.

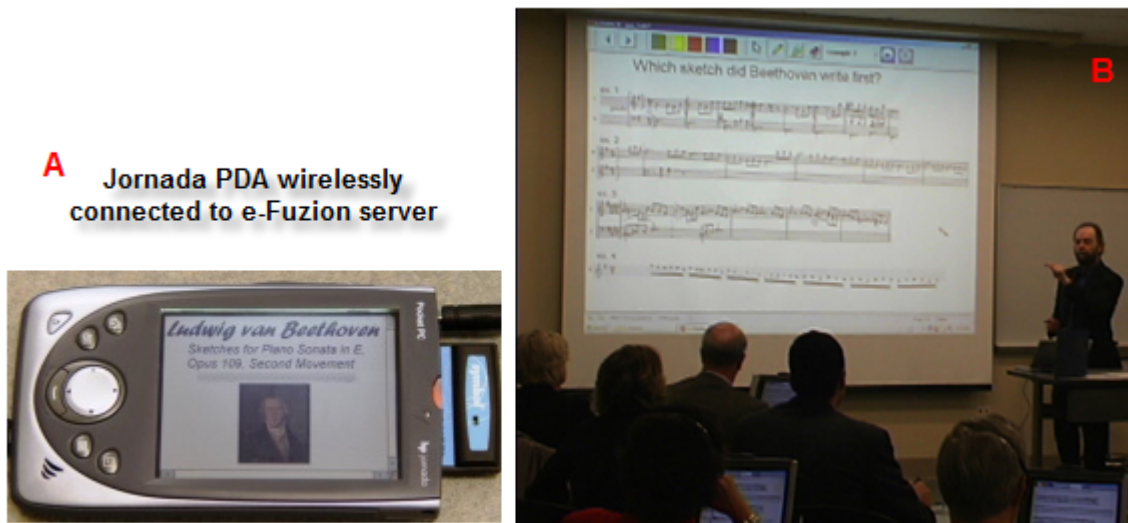


Figure 3.26: A) The presentation as received on a wireless PDA B) Professor Garnett of the Music Department teaches about Ludwig von Beethoven’s creative compositional process

Microsoft’s “rich ink” is unique to Tablet PC’s, it cannot be easily integrated to non-Tablet PC devices. The network architecture used in the Siebel Center grand opening demo abstracted away the intricacies of the rich ink and generalized it into a form accessible to web-enabled devices. In order to encapsulate the ink stroke data into a portable format, we first packaged the properties of the ink into an XML-formatted message. By embedding the ink stroke data into an XML message, we were able to attach meta-data tags to preserve context information including pressure sensitivity, vector graphics data, notes, and information such as the date, activity, and slide number. Once the XML-based message was constructed, the system signed, encrypted and assigned a reference number to uniquely identify the stroke.

Clients in our system fell into two classes. The first-class clients were Tablet PCs, which support rich multimedia capability in the form of ink and video; the second-class devices were web terminals, PDAs and smart phones. These devices were not capable of running custom software to support ink natively. To display ink information on such devices we used a custom ink renderer that utilized the information stored in the XML message. In order to support thin clients⁴ the distribution server⁵ used the information embedded in the XML messages to reformat the ink strokes to fit the various screen size and orientations of the non Tablet PCs and PDAs. This usually resulted in the server’s creation of a bitmap representation of the ink strokes.

⁴A thin client is a client computer or client software in client-server architecture networks which depends primarily on the central server for processing activities, and mainly focuses on conveying input and output between the user and the remote server. Source: http://en.wikipedia.org/wiki/Thin_client.

⁵The primary role of the distribution server was to provide transparency to all connected devices. As such, the server reformatted instructor-created ink strokes to accommodate connected devices.

3.8 Conclusion

In this chapter we introduced the Tablet PC and discussed its potential and promise. Although there has been a great deal of research performed on pen-input devices of the past, there is much exploration yet to be done that leverages the improved inking technology, “rich ink”, of the Tablet PC. See the white paper based on information gathered at the Tablet PC and Computing Curriculum workshop (August 4, 2004) (www.cs.washington.edu/homes/anderson/tpc/tpcwhitepaper.doc).

Chapter 4

2002 Pilot Study

4.1 Introduction

Higher education has increasing obstacles to effective teaching—burgeoning class sizes with a concomitant diversity of learning styles and the increasing inability for traditional teaching tools to cope. The key element of successful teaching lies in the ability of an instructor to respond to students' needs and perceptions efficiently yet individually.

In a large classroom, effectively reaching each student on an individual level is not an easy task for an instructor. Large lecture halls with a capacity to hold 200 students are even more challenging (figure 4.1).

In such an environment, student interaction is limited by a number of factors: size of the class, the lecture format, reluctance to interrupt the lecturer to ask questions, and students' fear of embarrassment asking questions due to lack of knowledge. Student questions are not only difficult to pose but also difficult for the instructor to address. As a consequence, many students are reluctant to participate, content to be spectators. These negative factors grow more serious in direct proportion to the complexity of the ideas being presented.

Additionally, instructors find it difficult to assess students' understanding given the variety of student learning styles, the limits imposed by the spatial and acoustical environment, and the constant pressure on the instructor to stay on schedule. Accordingly, varied instructional means and strategies have been introduced and examined as a way to help instructors handle a large classroom. In particular, learning technology and multimedia have been recognized as effective tools to provide the greatest possible array of student input and interaction with the instructor, and to respond to students' miscellaneous needs and learning styles.

Interaction outside the classroom is also limited. Many courses at the University of Illinois have relied on electronic newsgroups, email, and more recently, content management systems (CMS) such as Moodle or Illinois Compass (powered by Blackboard) to distribute announcements and allow students to ask questions outside of fixed office hours. One novel design consideration of the system introduced in this chapter was



Figure 4.1: Large lecture classroom settings. The picture on the left was taken in our original Computer Science building (Digital Computing Laboratory room 1320) across the street from our new building, the Thomas Siebel Center for Computer Science room 1404 (pictured on the right).

the creation of a graphical newsgroup that acted in both a synchronous and asynchronous form. We wanted to build a system whose combined tools allowed a user to directly import excerpts of lecture slides, personal notes, and other correspondences directly into new newsgroup posts, providing the increased comprehension associated with visual learning. Ultimately, we wanted to create a system to function as a learning tool both inside and outside the classroom.

In the fall of 2001 we began considering ways in which software running on a handheld computer (see http://en.wikipedia.org/wiki/Handheld_device) (initially called i-Board and soon after changed to e-Fuzion) could become a potential remedy for this problem. We submitted a proposal to the Academy for Excellence in Engineering Education (AE³) (<http://ae3.cen.uiuc.edu/>) for funds to help us refine and augment this concept by exploring the development of a variety of software applications during the spring 2002 semester.

The idea of this tool was to enable questions from students to be relayed instantaneously to a teaching assistant (or course staff), who, in turn, could either answer the question directly or relay it to the instructor so that she could potentially incorporate the answer spontaneously into her lecture. In addition, this tool would allow students to view a live stream of the instructor's lecture content to augment their own notes on the lecture. For example, the software would allow students to copy instructor-created content from the live stream and add it to their own workspace. Furthermore, the interaction between student, TA, and instructor could be archived and made available (i.e. posted) on a “graphical” newsgroup for ubiquitous access.

Interactive polling (for the remainder of this chapter we will use the term poll as a synonym for quiz), via this same application, could help the instructor gauge the level of understanding among all of the students in



Figure 4.2: Sony VAIO® Slimtop™ Pen Tablet PC model PCV-LX900

the class, information that could be of immediate and future benefit. We eagerly began development of this system, and the focus of our work became the implementation and refinement of the learning environment. Ultimately, we aimed at designing a presentation system that could support a pervasive student-centered learning environment.

4.2 The e-Fuzion System

Our classroom presentation and instruction system, which we called e-Fuzion, is described in section 4.2.2.

4.2.1 Motivation and need

The first e-Fuzion system was motivated by a need to improve the Microsoft® NetMeeting whiteboard application. In 1999, well before the debut of the Tablet PC, the Department of Computer Science at the University of Illinois at Urbana-Champaign began experimenting with pen-based computing platforms. Two types of interaction became of particular interest: as a presentation tool for instructors in a one-to-many limited interaction setting (see chapter 5); and as a vehicle for engaging students in the lecture class by shifting their role from spectator to collaborator.

In conjunction with the Department's growing Internet distance learning program, a number of lecturers began experimenting with technology by moving away from the traditional marker-based whiteboard/chalkboard at the front of a large lecture hall, to an electronic whiteboard available in Microsoft's NetMeeting. In 2001, Sony introduced the VAIO® Slimtop™ Pen Tablet PC, model, PCV-LX900 (figure 4.2), the first ever pen based desktop computing device¹.

¹Although called a "Tablet PC", the Sony Slimtop™ predated the release of Windows XP Tablet PC Edition in November of 2002

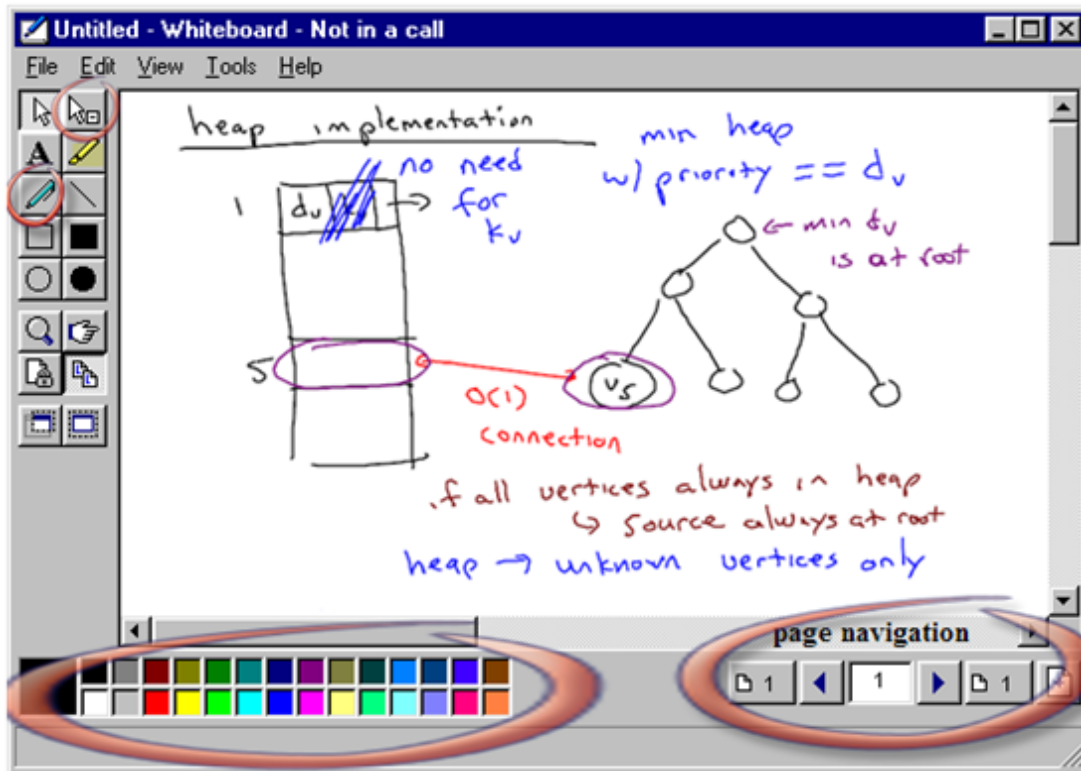


Figure 4.3: The Microsoft® NetMeeting application used by Jason Zych with a wacom pen-enabled input device

While there were a number of advantages to integrating this technology into the lectures, many limitations of the NetMeeting whiteboard soon became apparent, especially after we began to understand the potential power of pen-input devices. The toolbar icons inscribed in figure 4.3 were the only tools used during lecture; the pen tool, the select-erasure tool, the color tool bar, and the page navigation tools supporting the insertion, removal, and navigation to pages within the document. Although the instructor was using a device which incorporated an active digitizer (see section 3.3.1), the sampling rate of the user input was substantially reduced since the NetMeeting Whiteboard application was not designed to utilize the active digitizer. This is a feature that would be introduced by Microsoft® with the advent of the Tablet PC in November 2002. One by-product of the slower sampling rate is the sloppiness (non-smoothing) of the pen strokes.

One limitation of the NetMeeting whiteboard was the inability to *import* PowerPoint presentations. Instructors choosing to use prepared PowerPoint presentations in the whiteboard application needed to copy and paste each page of their original PPT presentation manually (by inserting a blank page in the NetMeeting whiteboard, selecting all the contents of the current PowerPoint slide, copying to buffer the selected contents, and finally pasting these contents onto the current whiteboard page) into the whiteboard application. While the transfer of static PowerPoint presentations into the NetMeeting whiteboard was

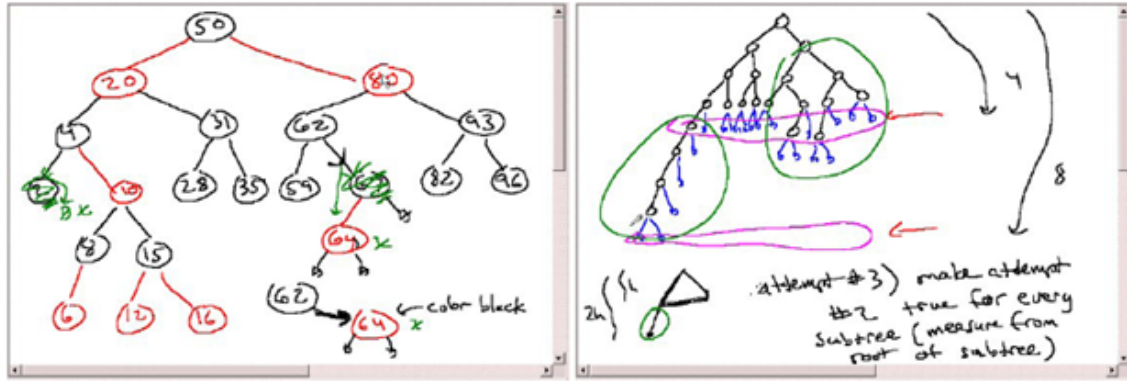


Figure 4.4: An excerpt from lecture notes created in Microsoft® NetMeeting's whiteboard application

possible, we found that the NetMeeting files grew in excess of 50 megabytes. This was in part due to the conversion of PowerPoint presentations into a set of large image files. The large file size made it difficult for instructors to save and email their presentations.

A fundamental problem with writing from a blank slate was the overhead to create and draw diagrams such as data structures (i.e. stacks, trees, and lists) from scratch. Figure 4.4 illustrates an excerpt from lecture notes created with the NetMeeting whiteboard. The page on the left hand side of the figure represents a red-black tree and took the instructor approximately three minutes and twenty seconds to draw². The picture on the right hand side of the figure was created earlier in the same lecture, and represents an unbalanced tree motivating the need for red-black trees. Both of these drawings were created on the fly during a CS225–Data Structures and Software Principles lecture. Illustrations created on the fly could be subsequently incorporated into the lecture—without expending the time to redraw the images from scratch—by using the copy and paste tools available in the whiteboard application.

From a teaching perspective, there are times when requiring students to copy what the instructor is drawing (in this case data structures) is a useful exercise. The use of a projection system instead of the traditional blackboard positioned at the front of the room (or a transparency machine or ELMO document camera) solves the problem of the instructor obstructing the view while writing. However, this solution does not solve the problem students encounter while taking notes. This is an example where a copy and paste functionality available in a whiteboard application and importing/inserting a presentation with figures would be helpful for the instructor but perhaps unhelpful to a student. The fundamental problem is that there exists a gap of interactions between the way students take notes and the manner in which the instructor uses the electronic whiteboard (figure 4.5).

²The duration for completing the red-black tree drawing was determined by observing and timing a screen recording of the lecture.

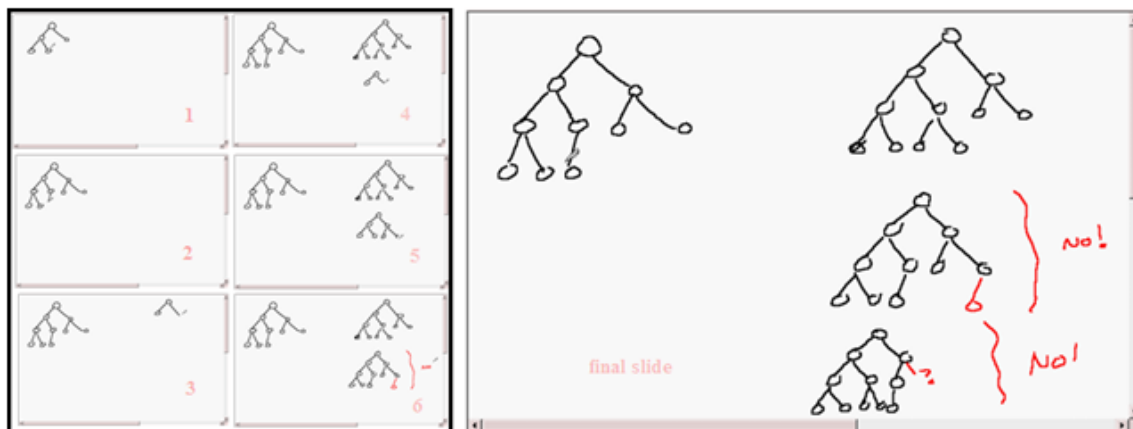


Figure 4.5: The image on the left illustrates a sequence of six frames derived from a video recording of a single slide within the Microsoft® NetMeeting whiteboard Application. The diagram on the right represents a snapshot of the same slide some time after slide six in the diagram on the left. One can imagine the amount of copying a student wanting to keep pace with the instructor would require.

4.2.2 e-Fuzion instructor edition v. 1.0

The instructor version of the software provided the tools available in the NetMeeting whiteboard application. The most significant improvement was that the application was based on a vector graphics library (see section 4.2.3). Aside from the scalability this design decision provided in a networked environment (section 4.2.5), the interface for inserting, resizing, and manipulating objects was greatly enhanced. A few implemented improvements included the support for importing PowerPoint slides, insertion of images, saving as PDF documents, and the addition of a separate area for taking notes with the keyboard. Figure 4.6 illustrates the added notes area for textual input.

The system also incorporated the use of a projector client responsible for the displaying of instructor drawing and annotations. This projector client component provided instructors with the freedom to wander about the room with their Fujitsu Stylistic 3500 (figure 4.7) un-tethered from the podium.

The ability for instructors to interact with the students in the lecture hall made the lecture more interactive; the instructor could ask for volunteers to provide solutions to questions posed by the instructor and the results were sent to the projector client and all the students in the class. The cost for this mobility was a latency of approximately 3-10 seconds between the time an instructor completed an annotation or drawing and the time it took to reach the projector display. One reason for this delay was related to issues of scalability in a 50 person classroom, both in the network implementation of our system (TCP/IP) and the general problems of network congestion. The network congestion was in part caused by sending all instructor slides to all the students in the class, at the same time. Since we were using the TCP/IP protocol for robustness and a unicast routing scheme, each student required a connection to the e-Fuzion server. In order to import

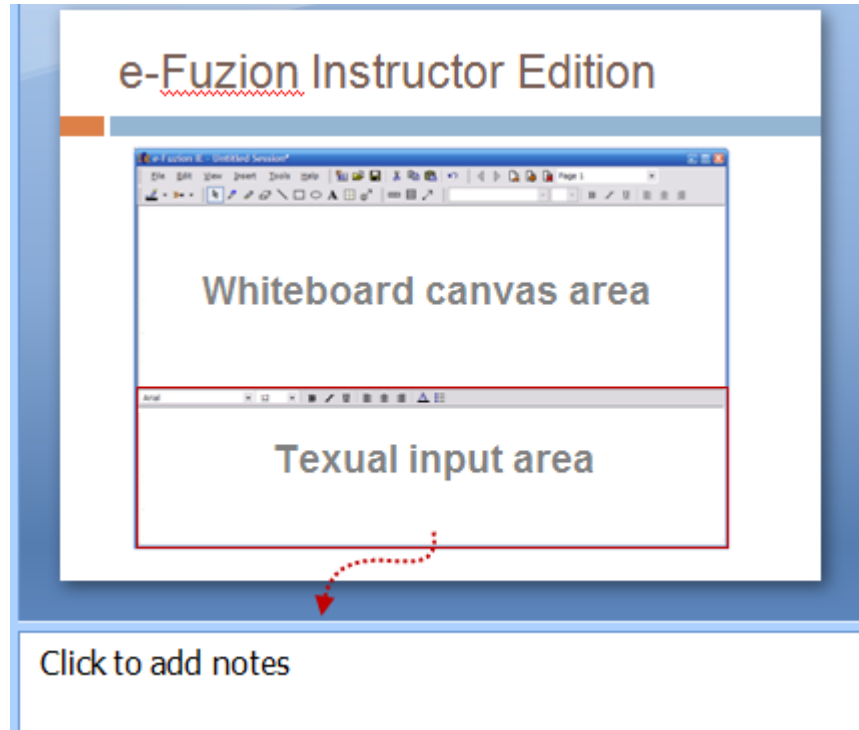


Figure 4.6: e-Fuzion instruction edition with textual input area

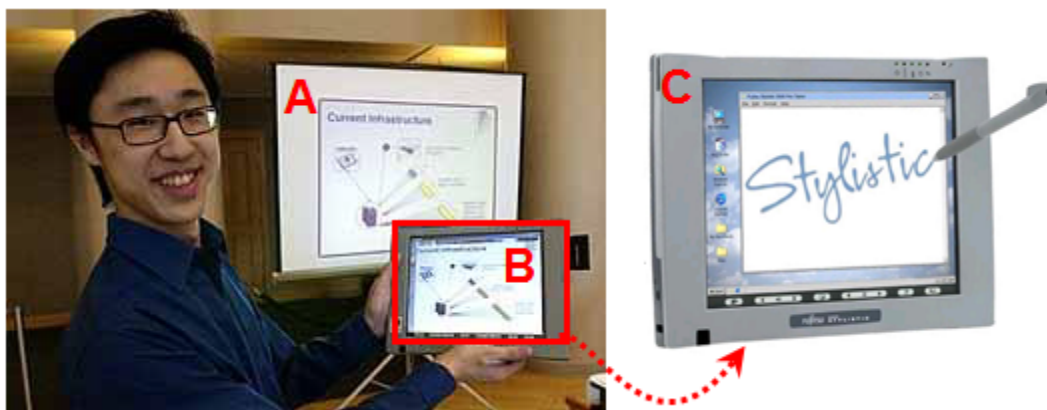


Figure 4.7: The image on the left was taken at the TeachIT 2000 technology conference on the campus of the University of Illinois at Urbana-Champaign. David Pan, one of the original developers of the e-Fuzion system is demonstrating how an instructor could walk around the room. A) Note that the image on the projector is the same image on the instructor machine (B). The device illustrated in part B is the Fujitsu Stylistic 3500 pen-input tablet.

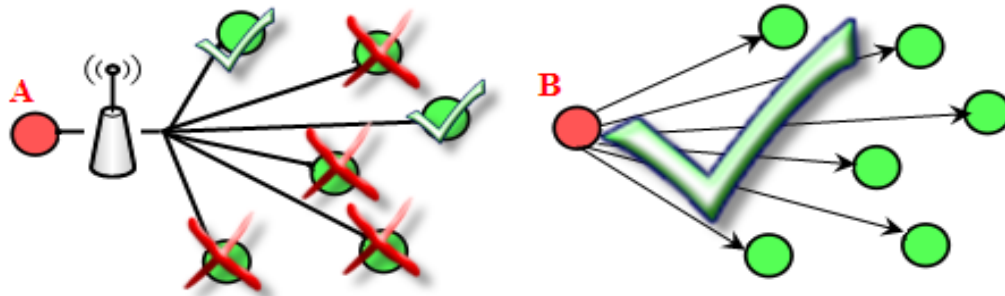


Figure 4.8: An illustration of two networking routing schemes.

a PowerPoint presentation into e-Fuzion, the system first converted the native PowerPoint vector-based data into a set of raster images (see section 4.2.3) and sent it to 50 client machines. Figure 4.8 illustrates two network routing schemes. The preferred method for sending all 50 clients the same data (in this case the e-Fuzion presentation) is illustrated in part A of the figure and illustrates a broadcast scheme. Broadcast refers to the transmission of messages that will theoretically be received by every client (green circles) on the network. However, there is no guarantee that all clients will receive the message contents—as is indicated by ✓ on the nodes successfully receiving the message—and for this reason, we could not use broadcast. The more common and “safer” networking scheme is called unicast and sends messages to single clients, one at a time. The disadvantage of unicast is that a connection from the server to each client is required. In a classroom of 50 students the network performance can become congested and result in increased latency.

4.2.3 Raster images and vector graphics

While instructor drawings using the vector graphics library were minimal in size when transporting messages over the network (i.e. when sending instructor created notes to the class), a presentation consisting of a large set of raster images could easily bog down the network. To understand the importance and significance of the vector graphics library, it is necessary to understand the difference between a raster image and a vector graphic. Figure 4.9 illustrates a comparison of a portion of a logo in both raster image format (A) and in vector graphic format (B). While the difference in picture quality (as a result of scaling the image) is apparent—the upper image has degraded while the lower remains unchanged—the individual file sizes of each are vastly different.

Loosely defined, a raster image is a data structure representing a rectangular grid of pixels of color. The total number of pixels (also known as the resolution) and the amount of information stored in each pixel determine the quality of the raster image; the higher the quality the larger the file size. Vector graphics use geometrical primitives based on mathematical equations to represent images. This minimal amount

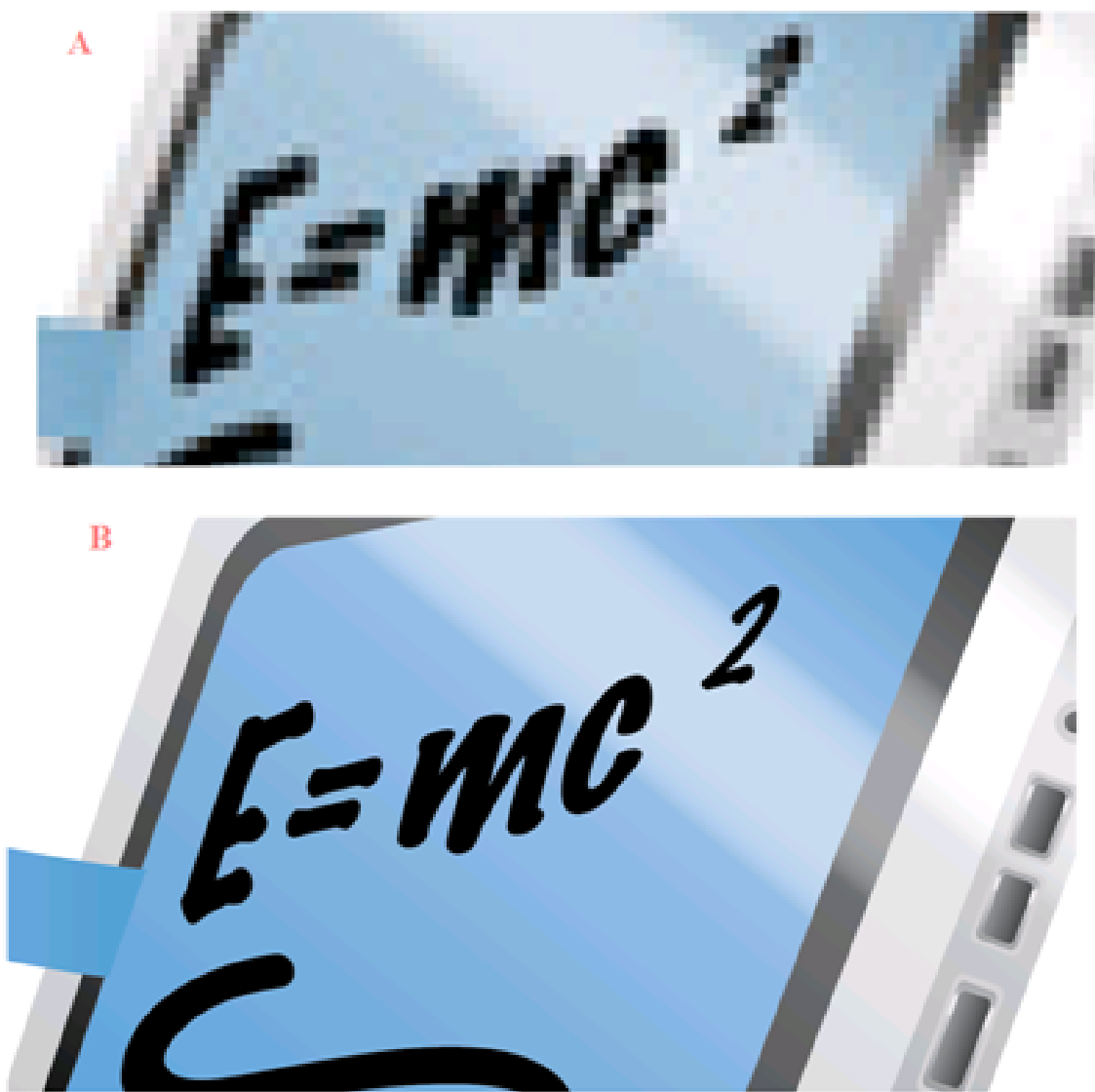


Figure 4.9: A) Raster image of a corner of the NuPaper® logo zoomed in to 1200% magnification. B) Vector graphic of the same sample zoomed in to 4800% magnification. The quality of the raster image is degraded (pixilated) as the magnification level increases while the quality of vector graphic is unchanged.



Figure 4.10: Handheld computing devices A) HP Jornada 720 Handheld PC. B) HPW-630ET Handheld Computer.

of information representing vector graphics translates into much smaller file sizes compared to large raster images. (More detail on the vector graphics library is given in ppendix C.2).

4.2.4 Fall 2001 and spring 2002 semester experiment

Increasing class sizes and the growing popularity of distance learning are making it more and more difficult for instructors to maintain the level of two-way communication with their students that is required to maximize student learning.

In an e-Fuzion learning environment, all participants in a course: instructors, students, and teaching assistants, have access to computing devices both inside and outside of the lecture hall. In the semesters leading up to the summer study we experimented with a number of small mobile pen-input devices in CS300–Data Structures and Software Principles for Non-majors, a class of approximately 40 students. The two devices we deployed were made available by the Department of Computer Science and from funds received from the AE³ proposal (figure 4.10).

In the fall 2001 semester we deployed 40 Jornada handheld PCs to test the usability of a networked graphical newsgroup. The Jornada 720 contained 32 MB of RAM, a Compact Flash slot, a PC card slot (for a network card), 640x240 16-bit display, and a 206 MHz StrongARM CPU. It ran under Windows CE 3.0. Although the HP Jornada had a stylus for input, the small screen size and passive display made the pen difficult to use within the application. Still, students were able to manipulate the stylus to use the few drawing tools we provided in the interface. As a result of integrating network cards into the units, the battery charge quickly diminished requiring power adapters to keep the units powered (even after requiring that the students charge their units overnight). We also experienced problems with acquiring IP addresses from the DHCP server and occasional network (802.11b) latency. This problem was resolved by the network administrators increasing the number of DHCP address allocations. The problems associated with the low sampling rate of the passive digitizer (see section 3.3.1) on the HP units made it difficult for students to use

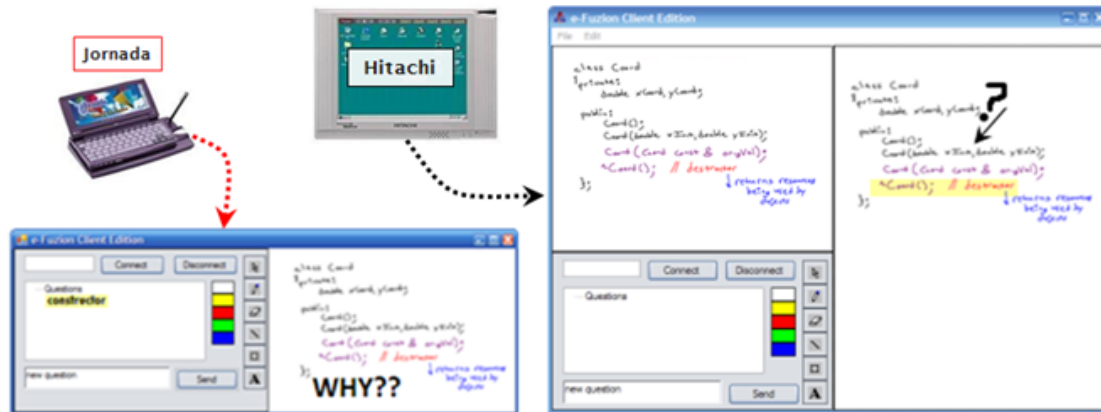


Figure 4.11: e-Fuzion client application interfaces for the fall 2001 and spring 2002 semesters

the drawing tools provided by the client application.

In the spring 2002 semester we deployed the Hitachi tablet devices which provided additional screen real estate. The Hitachi devices were more powerful, integrating the Hitachi SuperH™ SH-4 RISC microprocessor operating at 128 MHz, a larger display of 8.2" with a resolution of 640x480 and graphics accelerator chip; they were powered by Microsoft® Windows® CE Platform Builder version 2.12. Like the Jornada devices, they contained a PC port for a wireless network card and 32 MB of RAM. The most important lessons learned from these two semesters were the importance of screen real-estate and potential problems of network latency, issues that we would continue to deal with in the summer study. The small size of the devices required a creative user interface design to maximize the screen real-estate available in the Jornada and Hitachi mobile devices. Students were more inclined to use the keyboard to annotate instructor slides on the Jornada while more inclined to use the drawing tools on the Hitachi device. Figure 4.11 illustrates the e-Fuzion client application interface for the fall 2001 semester (on the left, text-based annotation using the keyboard) and the spring 2002 semester (on the right, pen-based annotation using the pen tool).

The form where students submitted their answers for quizzes supported only the use of buttons due to constraints of the screen size and user interface. Figure 4.12 illustrates an assessment sent synchronously to the class. An instructor could choose any subset of multiple-choice answers A/B/C/D/E, Yes or No, or True and False. Free response where students provided either textual input or ink annotations was not supported in this release of the software. It was difficult for students to use the pen-stylus on these smaller devices to draw "ink" annotations on top of instructor-created slides. While textual input was possible on the Jornada devices (albeit with some difficulty due to the small size of the keyboard), the Hitachi devices did not have a physical keyboard so students were limited to the stylus as their only means of input. The virtual keyboard integrated into the windows CE operating system took away a great deal of screen real-estate and as such

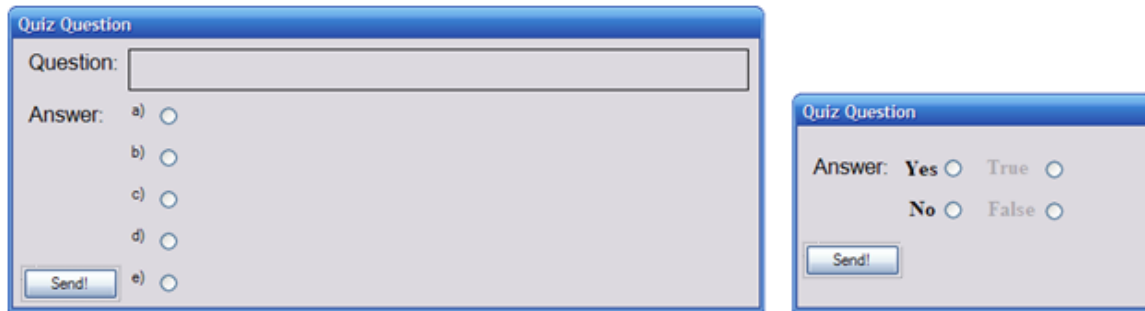


Figure 4.12: e-Fuzion on-the-fly poll

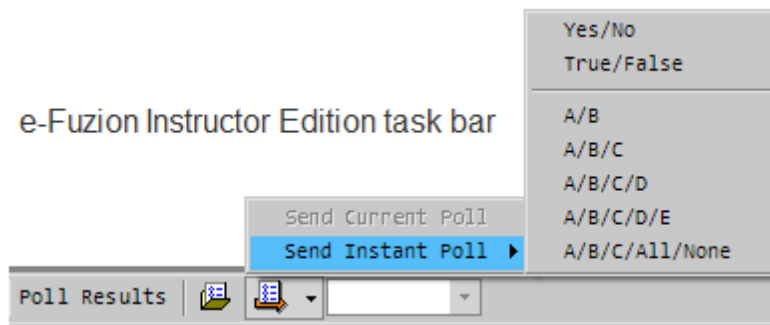


Figure 4.13: e-Fuzion instructor control for creating on-the-fly instant polls

was insufficient for use within the e-Fuzion client application.

Instant polls provided an instructor with a quick and easy way to send forced-choice questions (i.e. true or false, yes or no, any combination of A/B/C/D/E multiple choice) at the push of a button (figure 4.13). The instructor also had the option of adding a text-based question to the instant quiz. Many times, however, an instructor would just refer to their projected screens, having written the question on their own workspace. Upon selecting an option an empty skeletal question (such as those illustrated in figure 4.12) would be immediately sent to all clients currently logged into the system.

By the conclusion of our fall 2001 and spring 2002 semester experiments we had accumulated enough information including student feedback to refine the e-Fuzion application in preparation for our summer 2002 pilot study described in section 4.2.5 below.

4.2.5 Student and instructor in a networked classroom—summer 2002

In this study students used either a bulky traditional Dell laptop or a Fujitsu b-series laptop, lighter and substantially smaller with a passive digitizer and pen input (figure 4.14). A lottery was used to assign computers to students; the computers would be loaned (to take home) for the duration of the summer study. See appendix C.1 for the e-Fuzion documentation provided to the students in the pilot study. Upon running

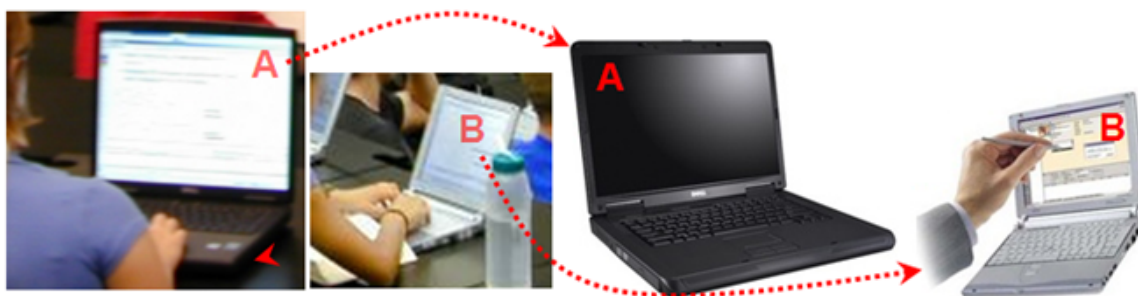


Figure 4.14: Laptops used in the summer 2002 pilot study

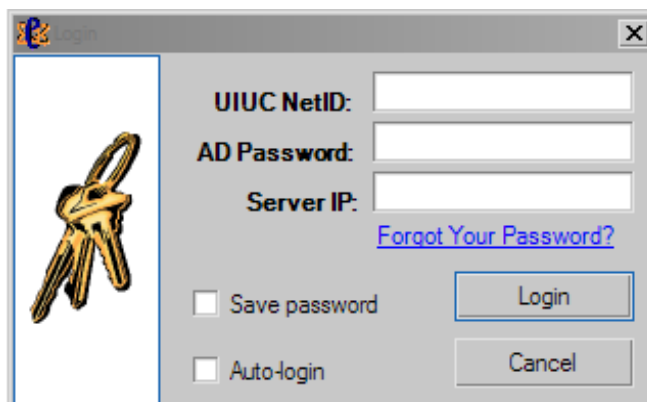


Figure 4.15: e-Fuzion login dialogue

the e-Fuzion application users were immediately prompted for their university network identification and active directory password in order to authenticate and login to the system (figure 4.15). After authentication³ students were prompted to select what course they wanted to join and had the option of saving that course (we only offered a single course in the 2002 summer semester) as their default course to bypass the course selector in future logins. Figure 4.16 illustrates the e-Fuzion options available for customizing a student's and instructor's session.

During lectures, instructors used the presentation tools available in the e-Fuzion Instructor Edition (figure 4.17) to annotate blank or prepared slides with notes and examples, which were immediately transmitted over the wireless network to student computing devices. Students could then use the same tools to add their own notes to the slides and save the annotated presentation for later reference. The system recorded usage for each user by logging user actions.

Aside from slide synchronization which provided students access to instructor slides in real-time, e-Fuzion integrated a graphical newsgroup that was available in-class (synchronous) and out-of-class (asynchronous). The e-Fuzion client edition (figure 4.18) allowed students to easily import lecture content, notes, and other

³AD password refers to active directory, a technology created by Microsoft that provides a variety of network services. See http://en.wikipedia.org/wiki/Active_Directory.

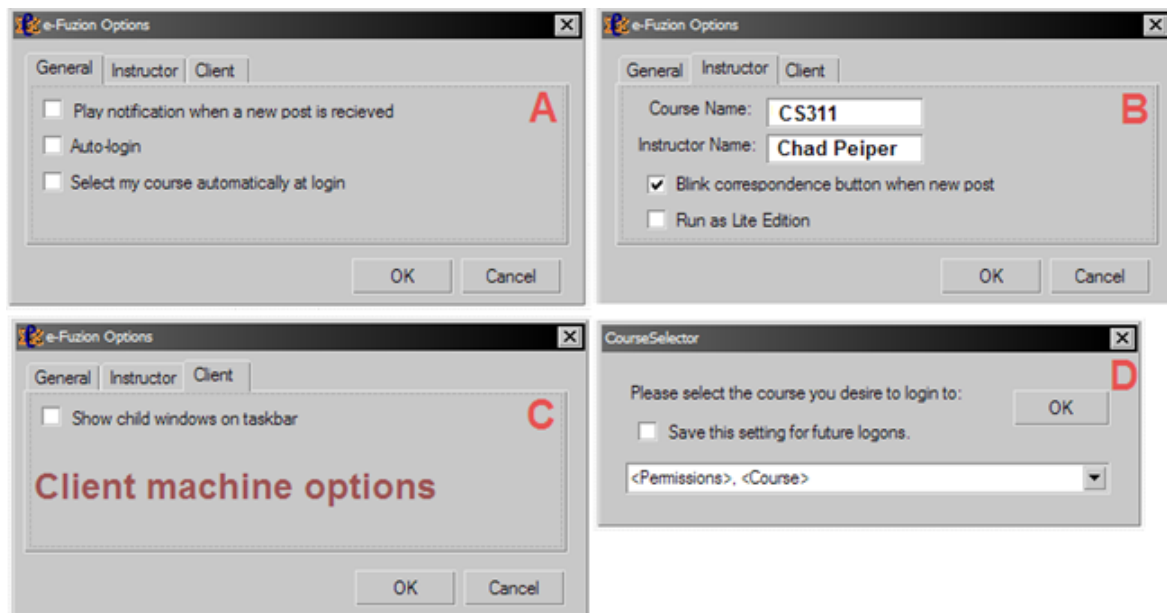


Figure 4.16: e-Fuzion options. A) Allows the user to specify whether they want to auto login with cached user credentials (the machines were not shared amongst users), automatically select course, and whether or not to display a notification when a message was posted to the newsgroup B) Default information for instructors running the application. The run as “lite edition” provided an instructor with a non-networked stand-alone application for presenting a lecture. C) Given the small amount of screen real-estate it was possible for students to maximize available space by hiding the taskbar. This option was an attempt to save space when polling and other e-Fuzion application windows popped up on a student’s machine. D) Course selector for all users authenticated in the system

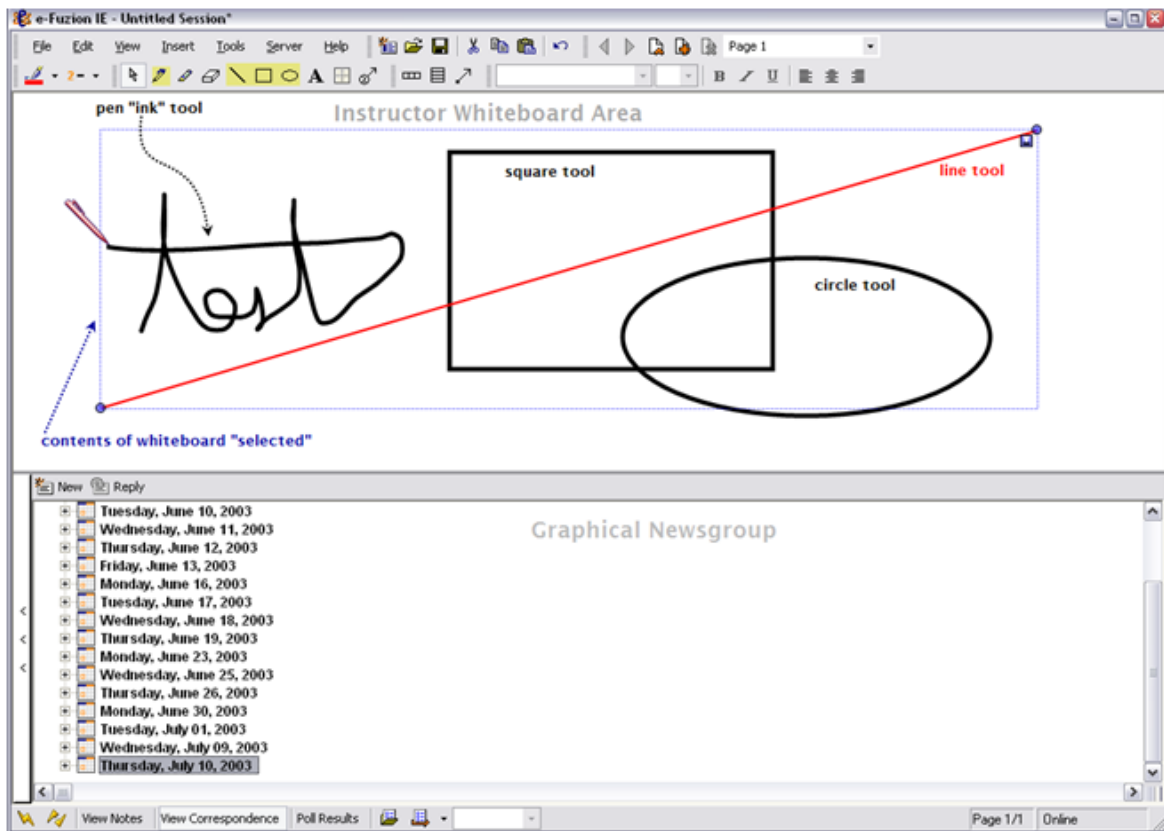


Figure 4.17: e-Fuzion instructor edition

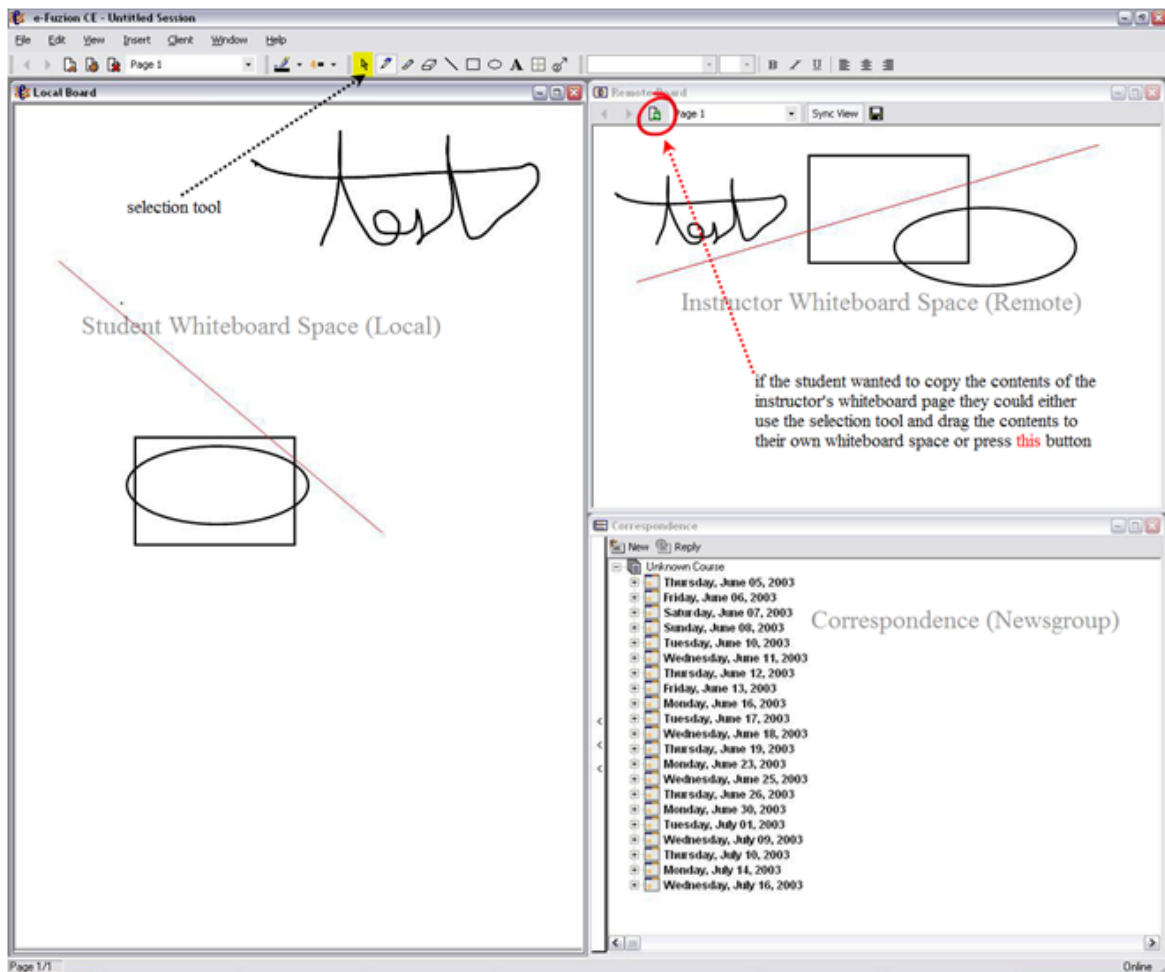


Figure 4.18: The e-Fuzion client edition. Students could copy content from selected instructor slides by either pressing a button (circled in red) on the remote board panel (upper right-hand corner) or by using the selection tool to select the desired contents on the remote board panel and drag them onto their Local Board (left-hand side of the screen).

correspondences directly into their new newsgroup posts, providing the increased comprehension associated with visual learning. Students were now able to articulate questions with simple annotations rather than spending the time to rethink and articulate their questions (from earlier class sessions) in ASCII text which was the only mode available in the university's UseNet [45] newsgroup system.

During a class session students could post questions to the newsgroup without interrupting the flow of the lecture. Teaching assistants would monitor the newsgroup during class, and could choose to answer questions as they occurred or signal the lecturer that a question had been asked requiring his or her attention. Teaching assistants no longer sat passively during the lectures and instead monitored student posts by adding a figure from the lecture to their response to a question, annotating the figure with additional explanation. Similarly, students could incorporate content from the lecture slides and their own notes into their questions.

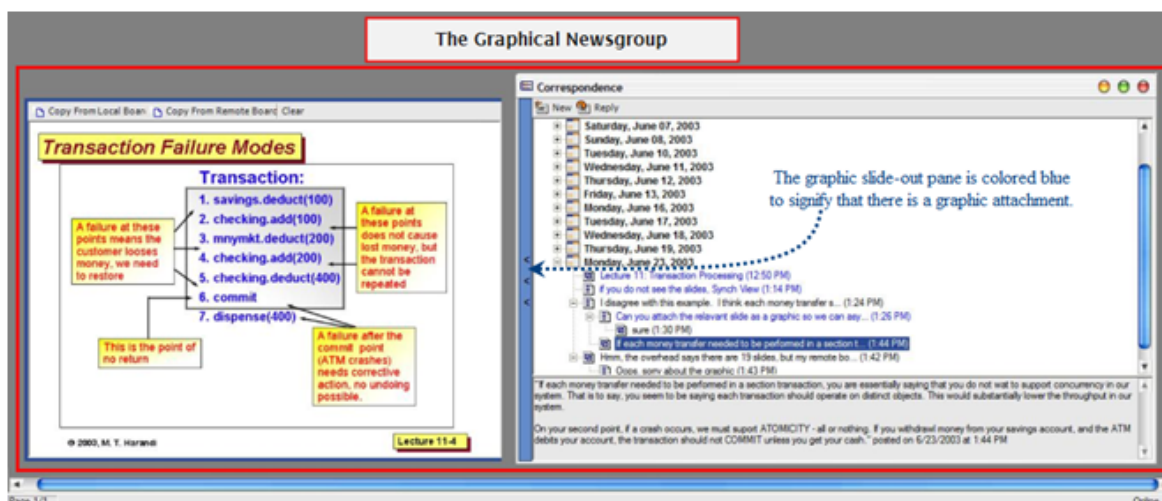


Figure 4.19: e-Fuzion graphical newsgroup

Figure 4.19 contains a snapshot of the graphical newsgroup. As displayed in the figure, the course staff posts were distinguished from anonymous student posts by coloring the subject lines of their posts blue. If a post contained an attached graphic (as illustrated in figure 4.19), the graphic pane would be colored blue rather than grey. Clicking on the colored pane expanded the graphic attachment for a user to view. This compacting of the attachments was an attempt to save precious screen real-estate on the client computers. The image contained on the graphic-pane was automatically scaled to the overall size of the Correspondence form. (The e-Fuzion documentation in appendix C.1 provides a more detailed description of the graphical newsgroup component.)

In addition, e-Fuzion provided facilities for incorporating active learning exercises into lectures, allowing students the opportunity to work with concepts covered in class and providing the instructor with instant feedback on student comprehension. Instructors could also create polls on-the-fly by copying textual or graphical data from their whiteboard and pasting them into the question pane. Figure 4.20 is an example of a text-based question requiring a graphical response in the form of pen strokes. The poll response form is pictured on the left hand of the figure (part A) and a student response is shown in part B. Questions requiring a graphic component would automatically be expanded (a mini-whiteboard pane would slide out similar to that of B in the figure). The sliding design feature was retained from our fall 2001 and spring 2002 study. The graphic pane (part B of the figure) was for free-type responses (it could also be used as small scratch paper area) using the pen tool.

An example of a poll integrating a graphic component in its question is shown in figure 4.21. Note the inclusion of the abovementioned graphic pane which in this example happens to contain the contents of a slide derived from the instructor's presentation (on the left-hand side of the figure and labeled A). In this

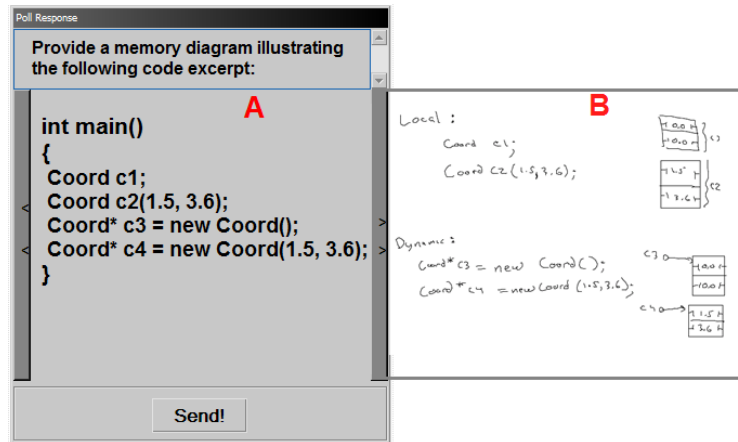


Figure 4.20: e-Fuzion interface for sending text-based question and requiring a graphical response

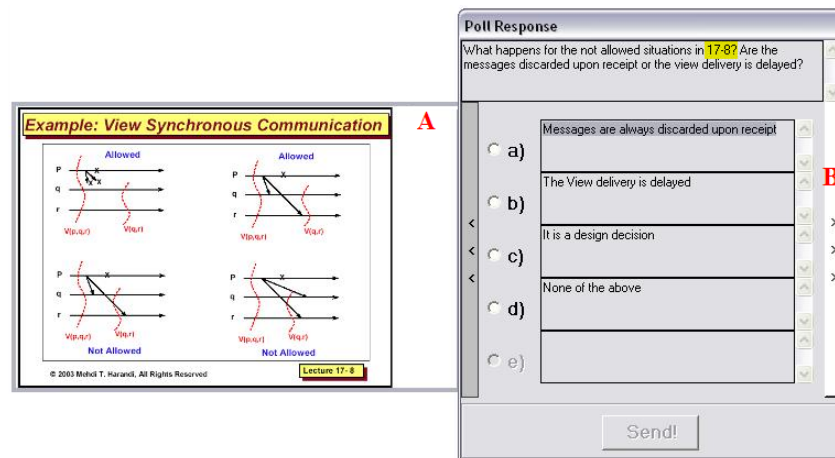


Figure 4.21: e-Fuzion interface for sending graphical illustrations and multiple-choice question response.

example, the instructor created a quiz ahead of time with four multiple-choice responses. Students were directed to review the contents of the graphical pane (the question, and in this case slide 8 of lecture 17) in order to answer the multiple-choice question. Note that students could utilize the graphical-response pane (on the right-hand side of the figure and labeled B) to work through the problem if they chose to do so.

The results of student polls were updated in real-time on the e-Fuzion IE (instructor edition) application. If the instructor wanted to display the results of the poll they could do so by clicking on the “Poll Results” tab on the status bar positioned at the bottom of the instructor’s e-Fuzion application (see part C positioned on the bottom of figure 4.22). The same hide/slide mechanism used to display attached graphical newsgroup content was also used in the result pane for polls on the instructor’s display. While the student responses were being received and aggregated in real-time, the necessary updates occurred in the percentage and number of student responses for the question. Figure 4.22 illustrates the e-Fuzion instructor poll response

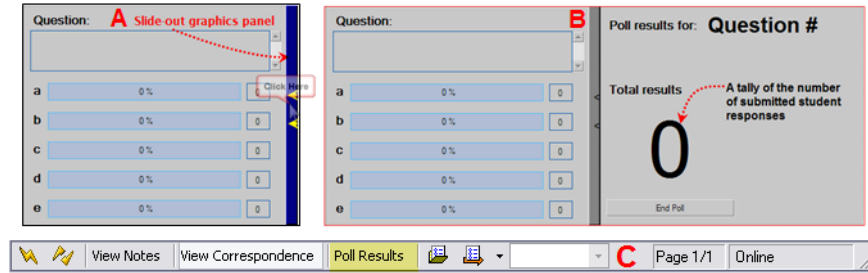


Figure 4.22: e-Fuzion poll results display

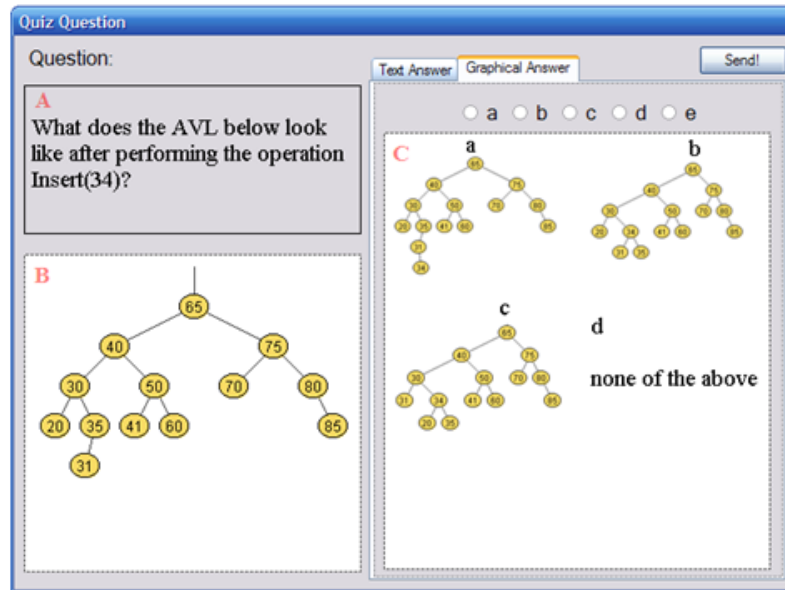


Figure 4.23: Illustration of textual, graphical, and multiple-choice poll

form with the number of tallied submissions hidden (part A) and the total results pane and end poll control expanded (part B).

An additional polling interface was added to take advantage of the larger screen real-estate available in the laptops used in the summer 2002 study. Figure 4.23 illustrates the combination of text-based question (part A), graphical component (part B), and a graphical answer choice component (part C).

The instructor also had the option of creating quizzes using the e-Fuzion quiz builder before the lecture (figure 4.24). The e-Fuzion poll builder supported the creation of prepared polls or quizzes. The Poll builder supported text-based, image-based, vector-based, or any combination of the three data types. The capture screen button allowed an instructor to directly import the contents on the current displayed page of the instructor's whiteboard into the quiz form.

As discussed earlier, we integrated into the system a vector graphics library (written by Patrick Bristow and detailed in appendix C.2) for the purposes of reducing drawing time and network overhead. Our intention

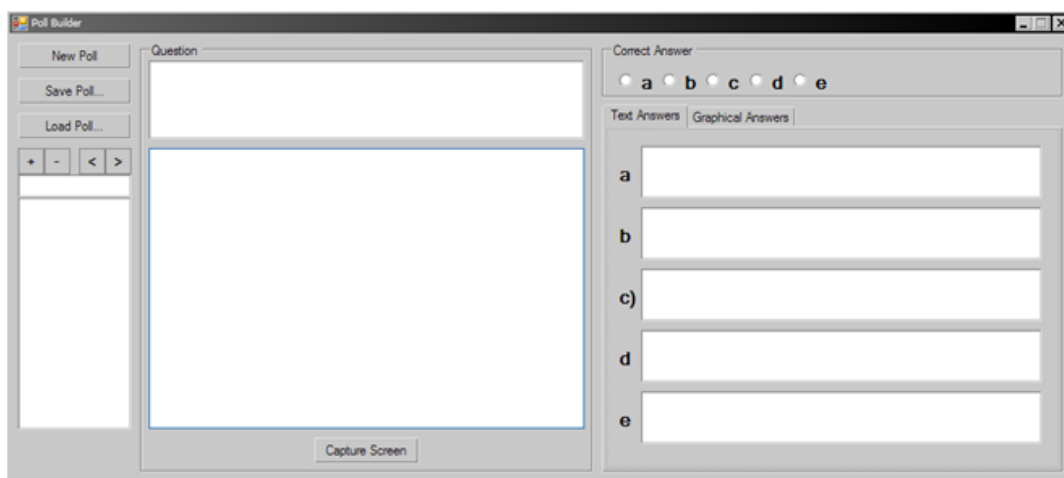


Figure 4.24: e-Fuzion poll builder

was to try to reduce the wireless bandwidth requirements when sending content from an instructor's machine to client machines. Unfortunately, our initial network model was not scalable; classes of 40 or more students experienced unacceptable delays in slide synchronization. Regardless, our system provided the flexibility of an instructor to walk about the room while lecturing with their computer in hand. An important attribute of our system was that the instructor was no longer tethered behind a podium with their device connected via VGA to the classroom projector. However, latency of instructor slides as they appeared on the classroom projector was a problem. Unless the instructor could somehow synchronize the delay of their written content with the delivery of their speech, instruction became more difficult to comprehend. What was needed was a wireless network protocol that could support the single broadcast of content to all clients—a multicast protocol—thus reducing the network congestion which was one cause of the problem. Unbeknownst to us at the time, slide synchronization and feedback in the form of polls was just one of the reasons for building a reliable network protocol. At the time, we were only concerned with the immediate sending of lecture slides and polls, and the ability for students to integrate that content into posts to our graphical newsgroup (figure 4.25).

Despite the potential of improving the learning environment and outcome in the classroom, however, no studies had been conducted to systematically test whether the e-Fuzion (referred to as "software" from this point on) had effectively met its goals of facilitating student learning and classroom interaction. The effectiveness of the e-Fuzion system, therefore, needed to be tested in actual instructional settings and environments.

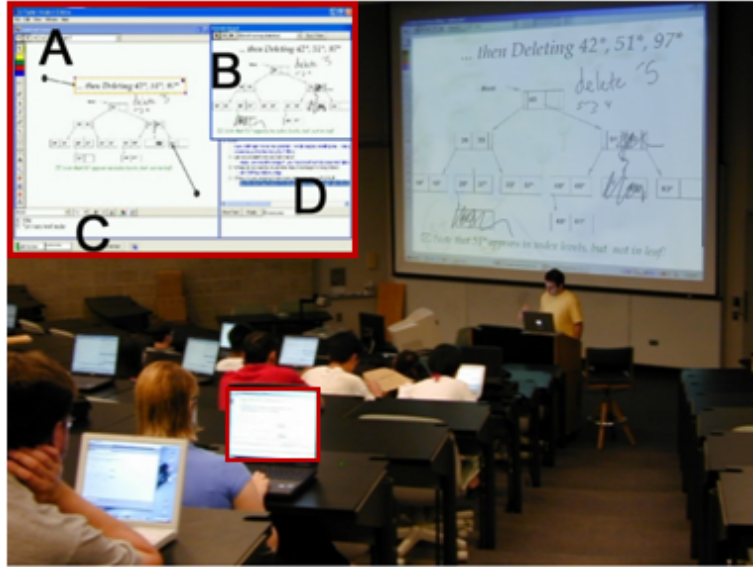


Figure 4.25: Student's view of e-Fuzion (upper left). Areas include: instructor's slides and notes (B), student's clipboard (A), student notes (C), and graphical news group (D)

4.3 Determining the Effectiveness of Learning Technology

According to Kozma [109, 110], the cognitively relevant characteristics of media, such as their technologies, symbol systems, and processing capabilities, can influence various aspects of the learning process. He contended that technology as learning media affects learning at every stage. Carter [26] also insisted that the question of whether media influence learning is neither trivial nor unimportant. Accordingly, it is anticipated that there is some kind of relationship between learning and media/technology, and further that media/technology affect learning.

Most studies examining the influence of learning technology on a student's learning have tried to demonstrate that students taught via an appropriate learning technology learn as much as, or even more than, learners taught without the assistance of such technology. These studies have focused on comparing learning scores between two groups of students under different learning environments, assuming that the two groups are identical at the outset.

In addition to student learning scores, the amount of interaction occurring during the course has been recognized as an element of successful teaching [131, 192] and so has students' perception of how easily, effectively, and clearly the learning technology could be used [10, 54]. There are a variety of different modes of interaction that take place in instructional settings: learner-instructor, learner-learner, learner-content [131], and learner-interface interaction [75]. With the increasing integration of technology in education, learner-interface interactions have become of increasing interest to researchers [75]. There are many studies

that report a significant impact of technology as a medium of instruction on modes of interactions, learning and overall teaching effectiveness [2, 64, 134, 191]. Another measurement used in the evaluation of technology is students' perception of the usability of learning software. Usability refers to the degree to which software can be effectively and easily used by humans in the performance of tasks [24, 161]. Dringus et al. [54] showed that students could work independently, enhance their overall performance, and control their own learning experience online when the learning technology was designed effectively. In particular, Anjaneyulu et al. [10] illustrated that a learner's perception of how easy a piece of software is to use and how useful they believe it to be has an influential role in the software's educational effectiveness and user perception. Through a detailed case study showing how usability testing could improve the quality of a computer-based course and facilitate an analysis of learning effectiveness, Crowther et al. [40] concluded that including usability testing as a part of the evaluation of software improves the quality and effectiveness of instruction mediated by computer.

Accordingly, our study investigated the effectiveness of e-Fuzion in terms of students' learning, their perception of its usability, and their interaction with the system. To establish the purpose of this study, the following questions were addressed:

1. Was there a difference in learning scores between students who took a course in 2002 with the use of the learning software and students who took the course in 2001 without the use of the software?
2. How did students perceive and interact with the e-Fuzion learning software?
3. Was there a relationship among students' learning scores, their interaction with the learning software, and their perception of the learning software?

4.4 Research Procedure and Design

A quasi-experimental design with a correlation component was used in this study. In general, studies where random assignment to treatment groups is not possible are considered to be "quasi-experimental." In the case of this study, the two groups were determined solely on the basis of student enrollments in one of two summer sessions of the same course in the department of Computer Science. This course was titled CS311-Introduction to Database Systems at the University of Illinois at Urbana-Champaign. The student group that took the course in 2001 was used as a control group while the student group in 2002 was used as an experimental group. In 2001, the course was given in a traditional manner, by contrast, all students who enrolled in the course in 2002 were given a laptop computer on which software was installed. They were required to bring the computer to every class and use it for taking notes or and asking questions during class lectures. In addition, students were encouraged to use the software and the computer for asking questions

outside the classroom as well as for communicating with other students in the class. Clark [36] criticized that the learning gains reported by many research studies exploring the effectiveness of learning media were in fact associated with changes to the method of teaching. To offset this potential criticism, the instructor who taught the class for this study used the same materials and exams with similar difficulty levels so that students' learning in both classes could be compared. The instructor fully understood that he should stay with the same teaching methods in order to facilitate correct outcomes for this study and to ensure the internal validity of this study.

To meet the purpose of this study, the two groups were compared in terms of learning scores. This study also explored students' perception toward the learning software and students' actual usage of and interaction with the software. Accordingly, this study included the following three dependent variables: learning scores, perception of the software, and interaction with the software.

Each variable was made up of several sub-dimensions. Exam scores and performance on problem sets were used as measurements of students' learning. The total score of four problem sets, the mid-term exam score, and the final-exam score that were used for the course grade were used as learning scores. Each score was recoded on a scale with a maximum of 100 points in order for the study's readers to easily identify which group scored highest and which exam showed the greatest difference in scores between the groups. The test scores were extracted from the instructor's grade books from both 2001 and 2002.

The students' demographic information (i.e. prior grade point average (GPA), the number of CS courses taken, their home college, and their status), obtained from university records, was used to ensure the similarity of the two groups.

Perception toward the software included the following sub-dimensions: barriers, efficiency, ease of use, and satisfaction. Thirty-seven 5-point Likert scaled items and three open-ended questions were administered at the end of the course to measure student perceptions of the software. Interaction with the software included the total number of logins into the software, the total number of quiz questions submitted, the total number of questions asked of the teaching assistant (TA) and other students during the class, and the total number of application tools used. The interaction logs were archived by the software.

Learning scores of students in the 2002 class and those in the 2001 class were compared to answer the first research question. The data from the survey questionnaire and interaction logs were analyzed for the second research question. For the third research question, correlation analysis was done to determine the relationship among the three variables.

Table 4.1: Students' GPA and the number of CS classes taken

	<i>N</i>	<i>M</i>	<i>SD</i>	<i>df</i>	<i>t</i>	<i>P</i>
GPA						
2002 Class	48	3.29	.51	101	.657	.513
2001 Class	55	3.35	.48			
The number of CS classes						
2002 Class	48	4.06	2.33	101	-1.36	.178
2001 Class	55	3.42	2.52			

Table 4.2: Students' demographic information

	2002 Class		2001 Class	
	<i>N</i>	%	<i>N</i>	%
College				
Engineering	24	50.0	23	42.6
LAS	10	20.8	10	16.7
Graduate	14	29.2	22	40.7
Total	48	100.0	55	100.0
Class				
Graduate	14	29.2	22	40.0
Undergraduate	34	70.8	33	60.0
Total	48	100.0	55	100.0

4.4.1 Subjects

The subjects of this study consisted of 55 students in 2001 and 48 students in 2002 who took the same course, were taught by the same instructor, used the same course materials, and were graded by the same criteria. However, the 55 students in 2001 were taught without the use of the software while the 48 students in 2002 were taught with the software.

The initial similarity of the two groups was determined through a review of demographic and performance information such as students' prior GPA, the number of Computer Science courses taken, their home department, and their year of study. The *T*-test statistic was employed to compare students' prior GPA and the number of CS classes taken. The results are presented in Table 4.1. The mean GPA of the 2002 class was 3.29 and that of the 2001 class was 3.35. On the average, students in the 2002 class had taken 4.06 CS classes and students in the 2001 class had taken 3.41 courses before they enrolled in CS311.

The two groups of students belonged to three colleges: Engineering, Liberal Arts & Sciences, and the Graduate College. Twenty-four students in the 2002 class (50.0%) and 23 students in the 2001 class (43.6%) were from the College of Engineering. Ten students in the 2002 class (20.8%) and 10 students in the 2001 class (16.7%) belonged to the College of Liberal Arts & Sciences. Graduate students comprised 14 (29.2%) and 22 (40.7%), respectively, of each group. Additional information is presented in Table 4.2.

Table 4.3: Sample questions to measure students' perceptions

Dimensions	Sample questions	Cronbach' α
Barriers	It took long time to learn how to use this software	0.96
	The TA seemed to be unfamiliar with the software	
	I disliked having to bring the computer to the classroom	
Efficiency	It was easy to learn with this software	0.95
	The software was helpful to discuss and chat with others	
	The software enhanced learning	
Ease of use	It was easy to remember how to locate information in this software	0.95
	I was able to access the materials without much difficulty through the software	
	I was confident in using this software	
Satisfaction	I felt a sense of satisfaction in this learning environment	0.97
	I would like to take another course using this software	
	Overall, I was satisfied with the software	

4.4.2 Instruments

As stated above, this study employed various methods, such as the measurement of student learning, tracking student interaction and use of the learning software, and a survey of student perception of the software. The exams and problem sets developed by the instructor were used as indicators of learning. The exams and problem sets were designed to have different maximum scores, but to meet the purpose of this study the exam scores were recoded on a scale with a maximum score of 100. In order to track student interaction and use of the software, the researchers utilized log files. A survey questionnaire developed by the researchers was used to measure students' perception of the software. A survey questionnaire, which consisted of 37 five-point Likert scaled items, included four sub-dimensions: barriers, efficiency, ease to use, and satisfaction. Barriers consisted of 10 items, efficiency was made up of nine items, usability had 10 items, and satisfaction included eight items. The sample items of each dimension are presented in Table 4.3. The reliability of each dimension as measured by Cronbach's α was 0.96, 0.95, 0.95, and 0.97, respectively, for the sub-dimensions (see table 4.3). Each dimension was defined as follows:

- Barriers: The users' feeling that barriers do not affect their learning and usage.
- Efficiency: The users' feeling that the software helps them to learn in a quick, effective, and economical manner.
- Ease-of-use: The users' feeling that the software could be easily learned and used.
- Satisfaction: The users' feeling good, comfortable, happy, or the opposite as a result of interaction with the learning software.

4.4.3 Data analysis

Data were analyzed by the following methods: multivariate analysis of variances (MANOVA), correlation analysis, and descriptive statistics. For the first research question, MANOVA was used to compare learning

of students who used and who did not use the software. MANOVA is a statistical technique used to determine whether there is mean difference among groups based on two or more dependent variables [71]. Because the learning indicators, the dependent variable of this study, consisted of three variables—problem sets, mid-term exam, and final exam—this study employed MANOVA to determine differences between the two groups.

Descriptive statistics were used to show the students' overall perception of the software and students' interaction and usage of the software. These statistics included the number of logins, the number of quiz questions answered, the number of questions asked in the classroom, and the number of tools used while in the lecture. The sum of item scores of each dimension was used as the indicator of students' perception. For convenience of comparing each dimension, the summed dimension score was transformed to a scale with the maximum score 100.

The Pearson Correlation analysis, which shows the relationship among variables, was employed to determine how closely related students' learning, perceptions, and interaction with the software were.

4.5 Results

4.5.1 Learning differences of students in control and experimental groups

Learning scores of students who took the class in 2002 and 2001, respectively, are presented in Table 4.4. Overall, students in the 2002 (the experimental group) class scored higher than students in the 2001 (the control group) class in terms of three learning scores: the problem sets, the mid-term examination, and the final examination. Specifically, the means of the total score of four problem sets were 91.39 ($SD = 12.78$) and 78.47 ($SD = 19.90$) in the 2002 class and the 2001 class, respectively. Students in the 2002 class also earned higher mid-term scores than did students in the 2001 class [71.93 ($SD = 13.05$) and 66.23 ($SD = 10.96$), respectively]. The final exam means were 65.03 ($SD = 14.14$) and 59.51 ($SD = 12.99$) in the 2002 class and the 2001 class, respectively.

In the test for homogeneity of variance—a diagnostic test for MANOVA—univariate tests for three variables were shown to be insignificant. Even though the overall equivalence test, Box's M , showed there was a significant difference between the levels of covariance, the significance level ($P = 0.036$) is deemed acceptable [71]. Therefore, MANOVA analysis proceeded. The significance of the multivariate test indicated group differences on the three learning scores at a 0.05 significance level (Hotelling's trace=0.168: $F = 5.53$, $P < 0.001$). According to Haie et al. [71] two-group comparison on multiple criteria is a special case of MANOVA and F -statistic is proper for the analysis. F values of each learning score were 14.89 for problem

Table 4.4: Learning of students in 2002 and 2001 classes

	<i>N</i>	<i>M</i>	<i>SD</i>	Difference
Problem sets				
2002 Class	48	91.39	12.78	12.93
2001 Class	55	78.47	19.90	
Mid-term				
2002 Class	48	71.93	13.05	5.70
2001 Class	55	66.23	10.96	
Final				
2002 Class	48	65.03	14.144	5.52
2001 Class	55	59.51	12.99	
Total				
2002 Class	48	228.35	31.87	24.14
2001 Class	55	204.21	35.05	

Table 4.5: Learning differences between 2002 and 2001 classes

<i>Variables</i>	<i>Between – groupMS</i>	<i>Within – groupMS</i>	<i>df</i>		<i>F</i>	<i>P</i>
			BETWEEN	WITHIN		
Problem Sets	15461.30	1038.65	1	101	14.89	0.000
Mid-term	559.00	96.48	1	101	5.79	0.018
Final	1050.22	246.61	1	101	4.26	0.042

sets, 5.79 for the mid-term exam, and 4.26 for the final exam (table 4.5). All of these F values were statistically significant at a 0.05 level of significance. When using the total learning score of the problem sets, the mid-term exam, and the final-exam, the effect size was 0.67 (their common SD of two groups was 35.56), which could be judged a mediumhigh effect [38].

4.5.2 Students' perception and usage of the software

Because each dimension of perceptions consisted of a different number of items, the total score of each was adjusted to the scale with a maximum score of 100. The mean of each dimension is shown in Table 4.6. The means of barrier, efficiency, usability, and satisfaction were 61.63 ($SD = 25.25$), 54.28 ($SD = 22.91$), 65.19 ($SD = 24.08$), and 61.01 ($SD = 27.80$) respectively. Students perceived usability of the software most positively and efficiency of the software least positively. In addition to Likert-scale items, three open-ended questions were asked: most liked things about the software, least liked things about the software, and suggestions for further improvement.

Seventeen students mostly liked that they could save and edit the lecture notes. They felt that it was convenient to write what the instructor said much quicker, easier to follow the lecture, and useful to go back and see what the instructor had said without interruption. Eight students liked the manner in which they could communicate and interact with the TA. They pointed out that they could ask questions of the

Table 4.6: Students' perception toward the learning software

	<i>N</i>	<i>Maximum</i>	<i>M</i>	<i>SD</i>
Barrier	48	100	61.63	25.25
Efficiency	48	100	54.28	22.91
Usability	48	100	65.19	24.08
Satisfaction	48	100	61.01	27.80
Total	48	400	242.10	93.21

TA without interrupting the lecture and that the communication was “very easy and useful.” About six students liked the software because they could more closely see what the instructor wrote. They seemed to feel they were somehow “closer to the instructor.” Additionally, students preferred the in-class quizzing functionality and the idea of being able to use the computer in class to increase their understanding of the lecture material.

On the other hand, students disliked the software for the following reasons: server connection/technical problems and insufficient functionality. Twenty-one students indicated server connection and technical problems as the least liked attributes experienced with the software. Ten students thought more refined functionality should be developed such as varied font style and graphic controls and added scroll bars. In addition, several students indicated other issues such as “the laptop computer was too heavy to bring to the class,” and “some of the windows interface elements were not quite as straightforward and easy to understand as they could have been.”

Based on the software experience, students suggested a number of improvements. Some students wanted to save the lecture notes in Portable Document Format (PDF) or other portable document types, while others wished to see the addition of scroll bars into the application and to have toolbars like other common applications. Quite a few students also indicated that a reliable network connection and more stable system should be guaranteed in future releases of the software. In addition, improved synchronization to the instructor's screen was another recommendation for improvement.


There was great deal of variance in terms of software use. Four in-class quizzes consisting of 43 questions were provided through the software, and the students answered an average of 16.58 questions, with the total number of questions answered ranging from 0 to 39. Thirty-nine out of 48 students (91.3%) responded to the quizzes. The mean number of questions asked to TA or other students during the class hours was 7.23, with numbers ranging from 0 to 81. The instructor expressed that this number of questions asked to TA and other students in this class seemed higher than in other classes that he had taught. Thirty-two students (66.7%) used this function during the class hours. The mean number of logins to the software was 14.69, and the maximum number was 23 logins. Two students never logged in to the software.

Table 4.7: The relationship among students' learning, perception, and the usage of the system

	1	2	3	4	5	6	7	8	9	10	11	12
1												
2	0.34**											
3	0.48**	0.61**										
4	0.91**	0.63**	0.77**									
5	0.20	-0.01	0.15	0.17								
6	0.20	-0.02	0.11	0.16	0.93**							
7	0.18	-0.15	-0.04	0.05	0.77**	0.76**						
8	0.23	0.06	0.08	0.18	0.77**	0.84**	0.88**					
9	0.18	-0.05	0.10	0.14	0.25	0.30*	0.11	0.17				
10	0.19	0.22	0.15	0.23	0.39**	0.30*	0.27	0.21	0.18			
11	0.12	-0.06	-0.03	0.05	0.14	0.10	0.09	0.11	0.28	0.31*		
12	0.28	0.09	0.04	0.20	0.44**	0.38**	0.34*	0.32*	0.33*	0.75**	0.24	

* $p < 0.05$ ** $p < 0.01$

1. Learning: the total score of problem sets.
2. Learning: mid-term examination score.
3. Learning: final examination score.
4. Learning: the total score of all tests.
5. Perception: barrier.
6. Perception: efficiency.
7. Perception: usability.
8. Perception: satisfaction.
9. System usage: the total number of tool use.
10. System usage: the total number of quiz questions answered.
11. System usage: the total number of questions asked of TA and other students.
12. System usage: the total number of logins to the system.

In terms of tool usage, there was a great deal of variance. The “select”, “pen”, and “screen copy” tools were the most used. They were used even a couple of thousand times by one student, while another never used these tools at all. Among 27 tools, nine tools were never used by any student. Six tools out of the nine, however, were ones used in teaching data structures. Among tools that students were expected to use, tools that were never used were “hand,” (a pointer device to emphasize content on one’s screen ) the “arrow,” (used to underscore objects on the whiteboard canvas \rightarrow) and “insert graphic” tools.

4.5.3 The relationship among students' learning, perception, and interaction

The relationship among students' learning, their perception of the software, and their interaction with the software was examined by the Pearson correlation analysis. The results are presented in Table 4.7. It was revealed that students' learning scores were not significantly related to students' perception of and their interaction with the software. However, students' perception of the software and their interaction were statistically significantly related to each other. To be more specific, barriers, the degree to which students feel that the software did not hinder their learning and usage, was related to the number of quiz questions answered ($r = 0.39$) and the total number of logins to the system ($r = 0.44$) out of four interaction variables.

Efficiency—the degree to which students felt that the software helped them to learn in a quick, effective, and efficient manner—was significantly related to all interaction dimensions except the number of questions asked to the Teaching Assistant and students ($r = 0.30, 0.30, 0.38$). Among interaction dimensions, the total number of logins into the system was related to all perception variables and other interaction variables except the total number of questions asked to the TA and students. This result implies that students’ feelings toward the software were related to their interaction with and actual use of the software, even though these were not directly related to their learning scores. In addition, the lack of association between perceptions of the software and measures of learning implies that these perceptions were not about learning bottlenecks so much as about varying expectations of software quality. This result suggests that usability may not in fact be a critical issue for learning in this context, but it still suggests potential usability improvements that could be made to the software.

4.6 Conclusion

This study examined the effectiveness of the learning software developed for improving learning in a classroom, particularly in a large classroom. In this study, which was based upon students who took a course in 2002 with the software and who took the course in 2001 without the use of the software, indicators of the software’s effectiveness were students’ learning scores, their in-class interaction, and their perception of the software and learning environment.

In terms of learning scores, students who used the learning software showed significantly higher learning scores than students who did not use it ($ES = 0.67$). Students who used and did not use the software had similar characteristics in terms of GPA, the number of CS courses taken, academic status, and colleges to which they belonged, and thus it could be assumed that they initially participated in the course with similar learning capabilities and knowledge. Therefore, it can be inferred that the learning software positively affected the learning of students taught in conjunction with the software.

Students’ perception of the software was investigated in terms of barriers, efficiency, usability, and satisfaction. Overall, students seemed to perceive the software positively. More specifically, students liked the software mostly for the following three reasons: (1) it helped them to save and edit the lecture notes; (2) they found it easy to communicate and interact with the instructor and TA via the software; and (3) they could feel closer to the instructor. However, quite a few students disliked the software because of an unstable server connection, technical problems, and insufficient functions. They also had some negative reactions because the computer was too heavy to bring to the classroom. Likewise students liked the instructional

aspects of the software while they disliked the problems caused by technical aspects of the software. If the technical problems that students had faced in the course could be fixed, the software could be recognized as an effective instructional tool to help teaching and learning in a large classroom setting. The instructor expressed his feeling toward the software during personal conversations with the researchers by mentioning that he was able to address students' questions quickly and efficiently due to the software. He mentioned that it allowed a faster turn-around in queries by students as well as modifications to the lectures based on students' feedback.

An analysis of students' interaction with the software reveals a great deal of variance in terms of the number of logins to the software, the number of quiz questions answered, the number of questions asked during lecture, and the overall usage of the software tools. In addition, students' perception of the software was positively related to their actual usage of it, even though neither the perception nor the actual usage was related to their learning scores. This result implies that it might be important to understand how students perceive software as it is being developed in order to enhance their actual usage of it. Without actual use of the software, it is hard to expect that the software would cause further impact on student learning.

This learning software was revealed to have great potential for improving students learning when it is appropriately used and free of technical problems. Therefore, it could be concluded that the learning software had a positive influence on student learning, even though other factors may have impacted the study's findings. More research is required to confirm this conclusion.

4.7 Implications and Limitation

The study presented in this chapter assisted in identifying students' experiences with learning software developed for potential use in a large classroom equipped with wireless technology, as is increasingly common in university settings. The study revealed that the software had great potential as an effective learning tool. Finding an effective learning tool and method is an important yet difficult task that an instructor should accomplish. Based on the information provided by this study, an instructor may attain a brief overview of a device to possibly employ in his/her classroom, employ the tool with reliable evidence of its effectiveness, and also learn how to best utilize the tool to facilitate learning.

This study was limited to data drawn from one engineering course with an enrollment of about 50 students. To generalize the results of this study to other college courses, more studies are required in varied instructional settings, including large lecture courses and those with varied populations. This further research may expand the potential of this software as an effective learning tool and also confirm the results

of this study beyond the setting of a computer science course.

This study explored student perceptions of the software usability. More usability studies, which can identify why some tools had never been used, why some tools had been used more than others, and how learners actually interacted with these tools, will be required for further adjustment and improvement of the software. These further studies will be able to provide more detailed information regarding how much and how well the software has been used.

One area of particular interest to us is the monitoring of student application use in real-time. Rather than relying on log files to present summary-based information in the form of statistics we would like the ability to evaluate the system while it is in use. By adding variables such as the measurement of student interaction with other applications (i.e. student is running a web browser and surfing the web) or observing the individual screens of the students' devices we can gain additional information. We refer to this type of information as "administrative" since it focuses on application management rather than the measurement of tool usage within a single application.

A revision of the software used in a Distributed System's class the following summer (2003) illustrated the administrative problems we could encounter if we did not monitor the students' interaction with the system. We never thought we would need to monitor student interaction in a university setting, nor did it occur to us that some of the problems we were experiencing with the wireless network (congestion) could be a result of student abuse. Unfortunately, we did not build into our system the ability to monitor individual client laptops during a class session. We thought monitoring student questions on the graphical newsgroup would be representative of the level of engagement in the classroom. It was not until we took a cursory glance at a random selection of the automated screen captures taken during the summer 2003 class sessions that we realized what was really going on during the lectures (figure 4.26).

The realization that student's were off task was not really surprising. What concerned us was that we had no indication or explanation why students were off task. Our system provided an infrastructure to share slides and allow real-time questions and answers, thereby encouraging discussion of the lecture content, but, students off task were completely out of the loop. The absence of feedback mechanisms to monitor the student interactions for their current state (i.e. bored, lost, confused) in our system necessitated a rethinking, once again, of the design of our system.

It would have been relatively easy to implement an application blocker to disallow students from running other applications on their computers. However, such a tool undermines the purpose of providing students with the technology in the first place, as a vehicle for learning. We were investigating how to better engage students in the learning process and in doing so lost perspective on what was happening in the real-time

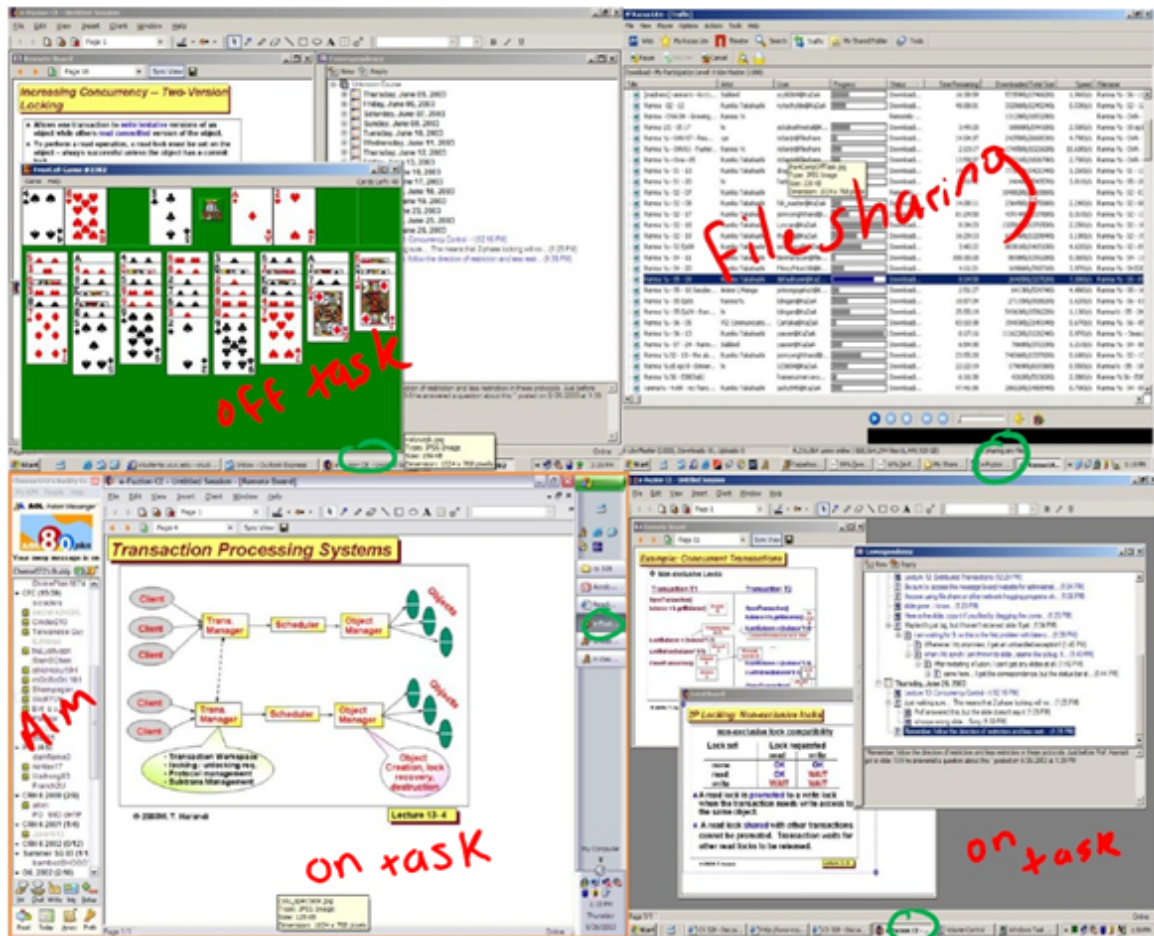


Figure 4.26: A second iteration of the laptop version of e-Fuzion was integrated into a summer course following our first study. Displayed in the figure above are snapshots of four student machines at random intervals during the course of the summer session. All but one student seems to be on task. Circled in green are the taskbar entries of the e-Fuzion client application (all of the students were running the application). However, there seem to be varying degrees of interaction with the lecture. The student represented in the screen in the upper right hand corner is completely abusing the bandwidth to download files, while the student beneath that is answering questions on the graphical newsgroup.

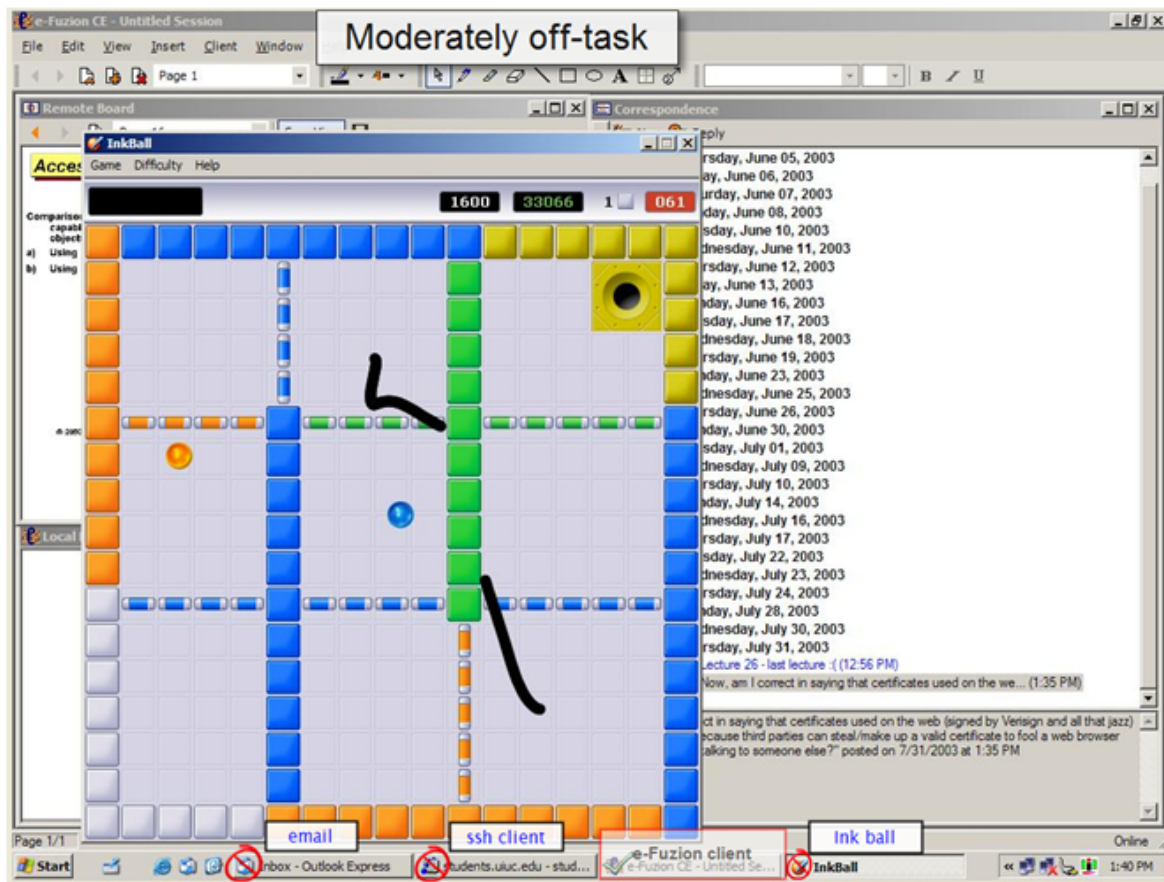


Figure 4.27: A student labeled moderately off-task. The student is running three applications not directly related to the in-class lecture. The ink ball is a game designed for the Tablet PC machines.

learning environment. If we had had a way of determining whether students were off task as defined by the running of other applications (i.e. in addition to the e-Fuzion client), the problem of managing these students would have been overwhelming. By reviewing randomly-taken screenshots of student machines in the summer 2003 Distributed System's class, we observed varying degrees of how on-task or off-task students were. Figure 4.27 is an example of a student running three applications. Something as simple as counting the number of running processes on student machines could have been more informative at the time. While this might seem far-fetched, take as a contrasting example figure 4.28 which illustrates a student who (by this simple heuristic) is deemed to be completely off task. This student's machine is clearly engaged in activities associated with personal computer-usage (surfing the web, emailing, a spreadsheet program and a number of instant messenger conversations) rather than the curriculum-based interaction that e-Fuzion tried to encourage.

An important lesson taken from this study was a common pitfall many early adopters of educational technologies experience: providing technology without a clear understanding of how it should be integrated

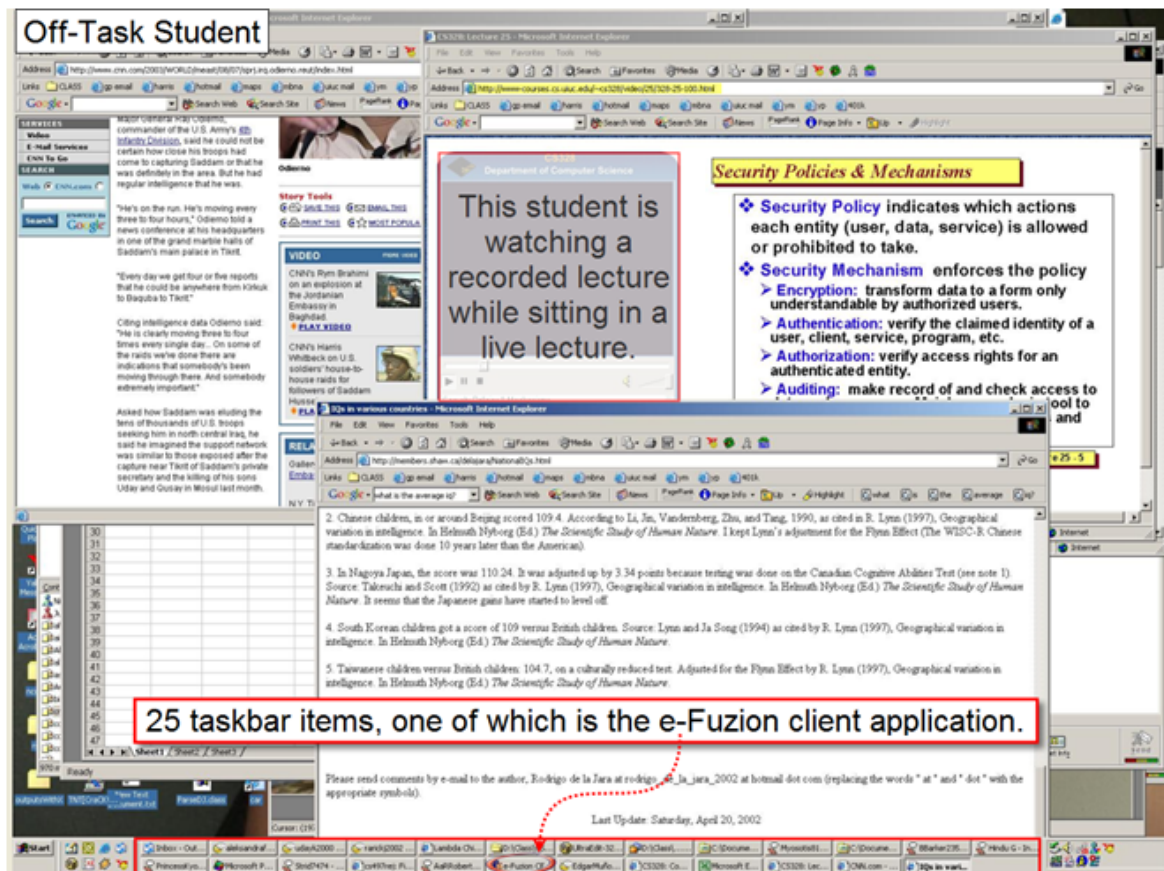


Figure 4.28: This student is completely (very) off-task. The student's workspace is populated with 25 windows/applications. An almost humorous observation is that this student is sitting in a live lecture watching a recorded lecture given earlier in the semester.

into a course curriculum provides a vehicle for students to get off task. What we wanted was to provide a tool to facilitate learning and student engagement in lectures. As a result, a monitoring system became a required component for a future system where every student in the room has a computing device. In chapter 7 of this thesis we present a preliminary dashboard design for presenting information based on user interactions recorded within our Tablet PC-based system.

Chapter 5

Instructor-only Usage

You don't understand anything until you
learn it more than one way

Marvin Minsky

5.1 Introduction

In this chapter we discuss how the standalone (non-networked) version of e-Fuzion was used by two instructors teaching two of our department's core undergraduate Computer Science courses: CS125–Introduction to Computer Science and CS225–Data Structures and Software Principles. We have several semesters of data from spring 2005 to spring 2007. We will describe how the design of our instructor-only system evolved over this two year period, highlighting a number of observations related to instructor tablet use. We present the results of student surveys that were taken for one of the instructors across three semesters. The majority of this chapter will be spent presenting an analysis of how one of the instructors used the system from a number of different perspectives.

5.2 Overview of Instructor Usage

Cinda Heeren and Jason Zych, two Lecturers in the Department of Computer Science responsible for teaching the introductory core courses, volunteered to use our whiteboard application in their classes. The average class size for courses offered during the fall and spring semesters was approximately 150 students. The summer semesters were much smaller in size with a maximum enrollment of 30 students. All the classes were taught in the large lecture hall in the Siebel Center room 1404.

Jason and Cinda had very different teaching styles. As an analysis of their individual usage across a number of semesters shows, Jason came to lecture and used the whiteboard as a blank slate, almost exclusively creating documents from scratch, while Cinda imported prepared presentations. In the latter case, annotations dominated the instructor's usage pattern since a majority of her slides contained images

created ahead of time.

There is a fine distinction between these two modes of interacting with a whiteboard application. The differences are in some part characterized by the meanings of the terms. Wikipedia defines annotation as “...extra information asserted with a particular point in a document or other piece of information.” American Heritage’s definition is “the act or process of furnishing critical commentary or explanatory notes.” The common factor is the adding of information associated with a particular point in a document. We contrast annotating with the act of writing. Writing refers to the creation of content that might later contain annotations. We will come back to the semantic and application usage differences in section 5.5.

5.3 Application Interface Design

Our first Tablet PC-based application was written as a standalone, non-networked application used solely as an augmented electronic whiteboard. This iteration of the software development process was a retreat from our previous version in that we had to start from scratch. This was in part due to the fact that the vector graphics library, a fundamental component of the previous system, was not designed to work with an active digitizer. All the drawing functionalities including the “inking,” shapes, and course-based modules were implemented by the vector graphics library. Still, the experience and problems we encountered with the network and user interface design in the previous system assisted in our initial design of this version. Feedback in the form of student surveys, recorded log files, and live observations collected from the summer 2002 study aided in the process of our development.

e-Fuzion 2.0 was designed for instructor use in large lecture halls with a projector (figure 5.1). Similar to the Microsoft® NetMeeting application, an instructor was tethered to the podium where the Tablet PC was physically connected to the projector system. Recall that in the previous system an instructor could walk freely about the room with their device in hand. This was not so in the newer Tablet PC version which in the beginning had no network integration. In the beginning stages of development, a majority of the work was spent learning the new inking API and deciding what features from the earlier system were required. Unfortunately, logging application interactions was not a feature in the first few releases of the system. Figure 5.1 illustrates the first version of our Tablet PC-based e-Fuzion system. This version of the system did not include scroll bars nor did it provide a mechanism to increase the size of the whiteboard canvas. Figure 5.2 illustrates how the contents of the whiteboard were scaled according to the size of the application window.

The integration of the Microsoft® Inking API necessitated a complete rewrite of the system. The more

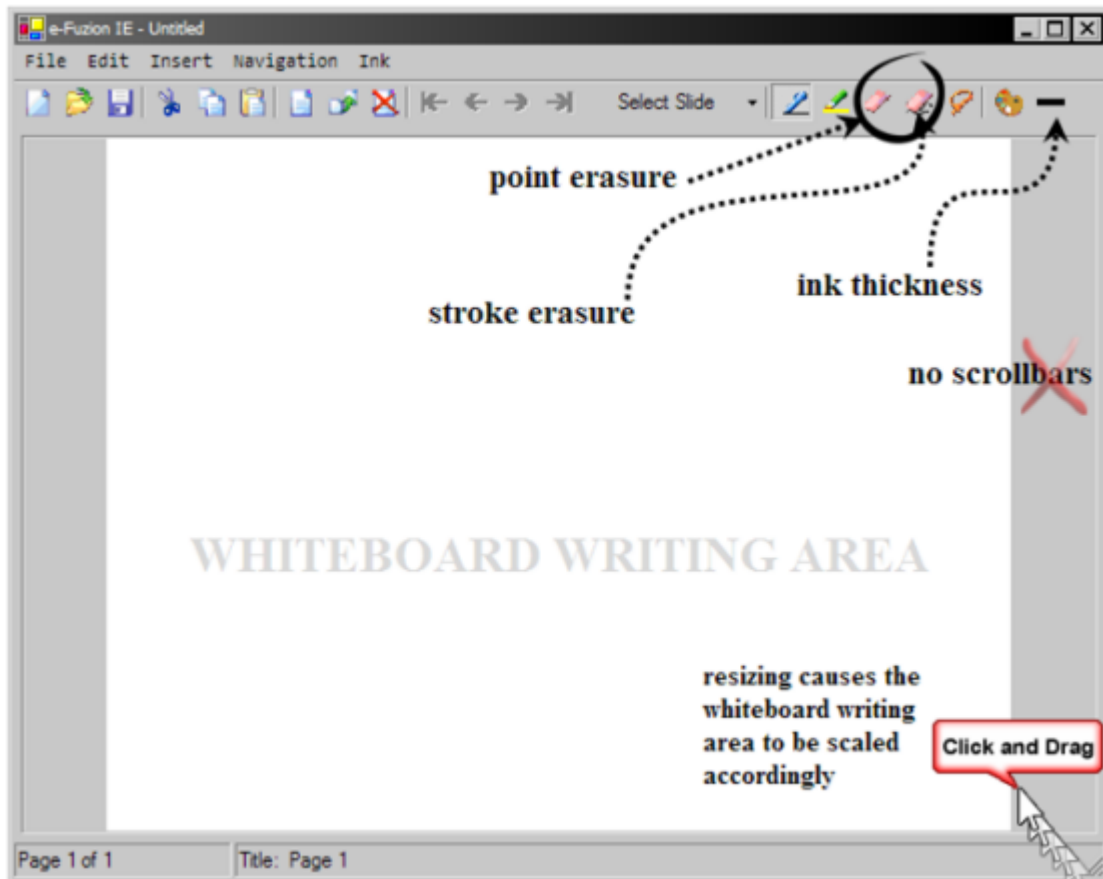


Figure 5.1: Pre-logging version of our e-Fuzion instructor edition. Extremely basic functionality using the Microsoft® Inking API. Note the two different types of erasures; the first models a real erasure in the splitting of strokes while the second erasure removes entire strokes rather than tangential points. Although this version did not have scrollbars, the entire whiteboard scaled according to the verall size of the canvas (see figure 5.2).

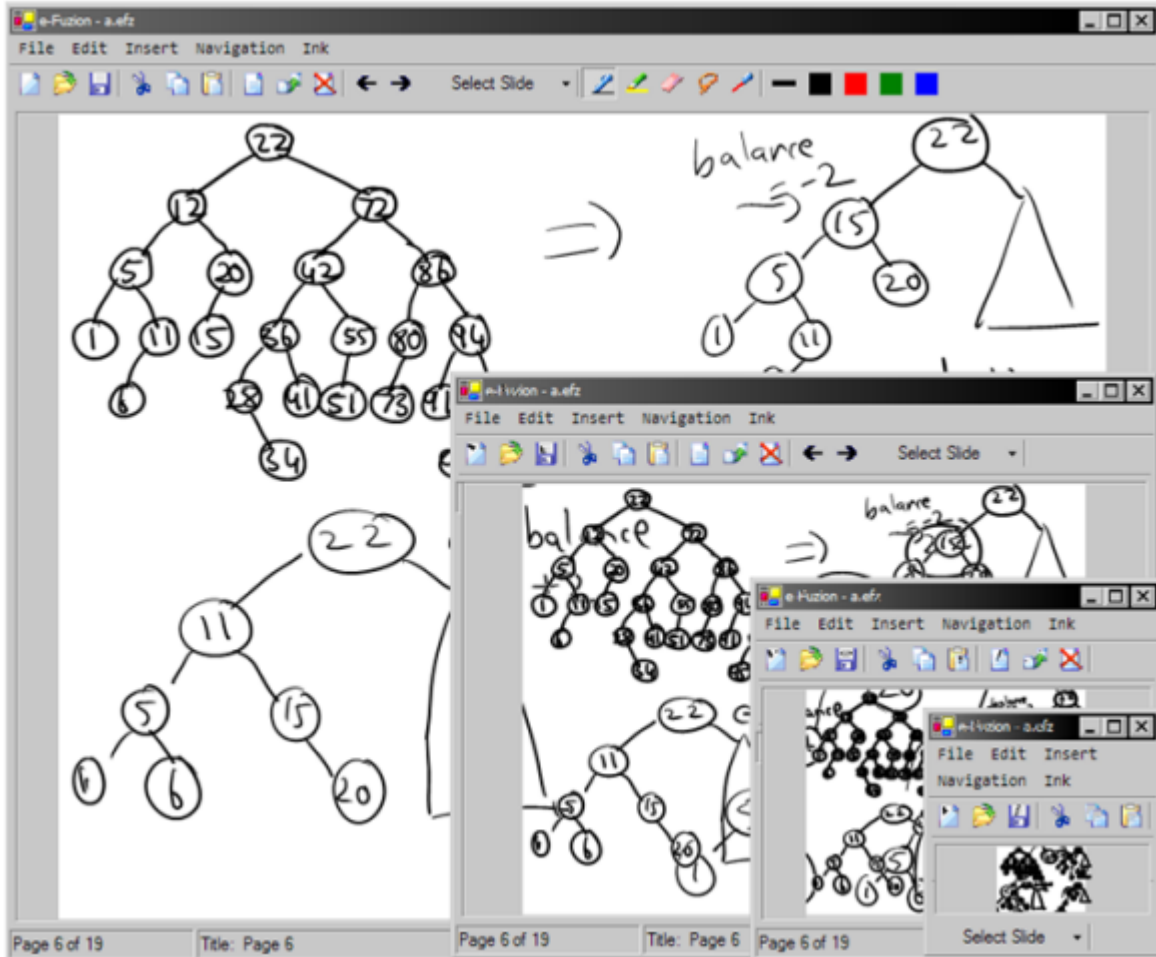


Figure 5.2: An illustration of our second release of e-Fuzion instructor edition. We added four colors requested by the instructor, taking away the color pallet which required two clicks for color selection. Similar to our laptop version in the summer 2002 study, we did not implement scrollbars to navigate a large canvas. Nor did we implement zooming in and out of the whiteboard area. The canvas was instead scaled according to the resizing of the application's window.

time we spent observing instructor usage patterns the more ideas we formulated for subsequent releases of the application. Still, it would take at least two years to come to the realization that less is more. Providing an instructor with a simple user interface that provides useful tools to teach should have been our underlying principle from the start. However, experimenting with the myriad of possibilities available in the rich inking technology made it difficult to do so.

Figure 5.3 provides more details on the functionalities available in our second e-Fuzion release. After about three months of distributing the application we started to receive requests from a number of instructors wanting to save native e-Fuzion (efz) documents to Portable Document Format so that they could make their presentations available to students after the lecture.

Additionally, instructors wanted to import PowerPoint presentations into the application. This release

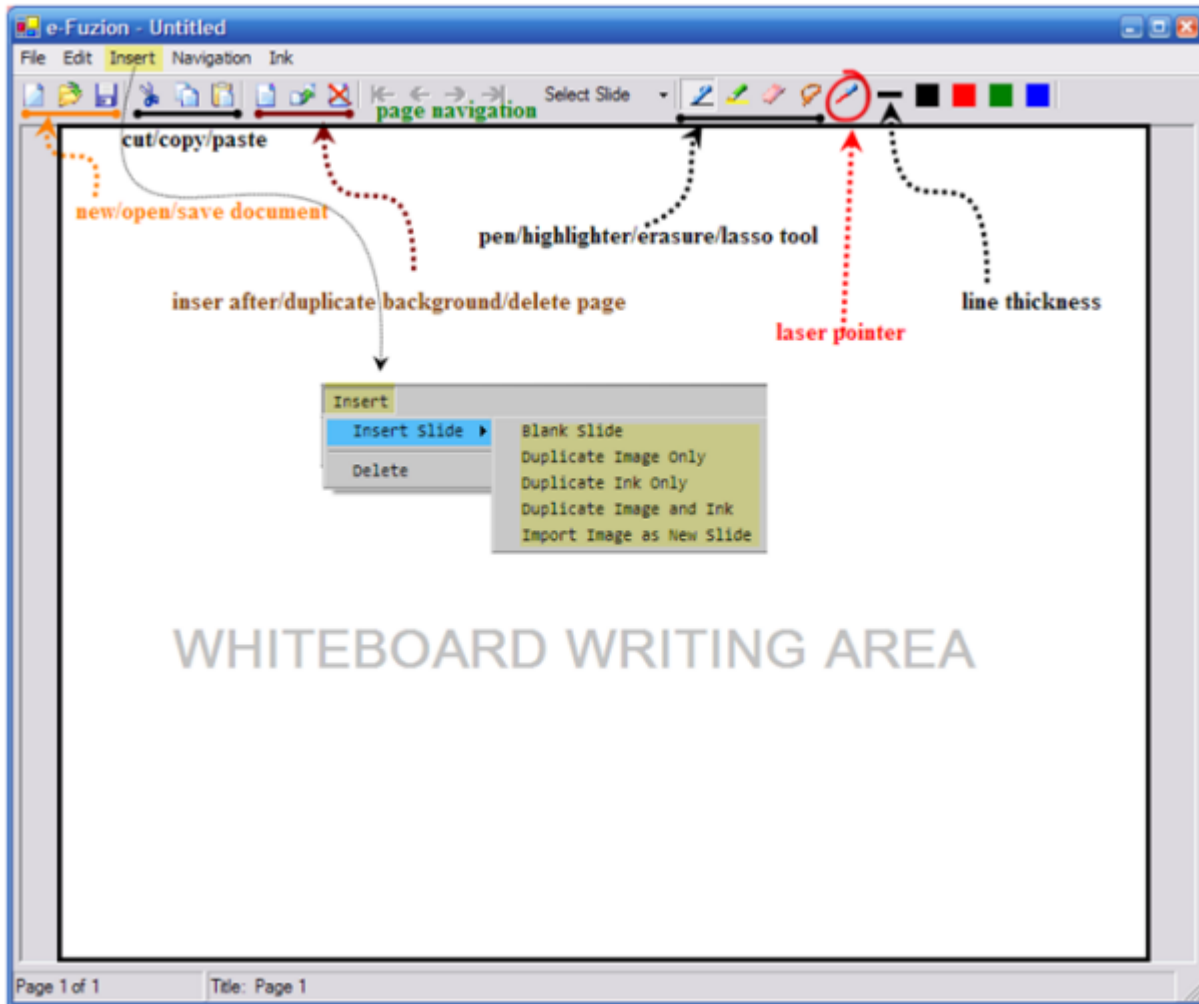


Figure 5.3: This interface provided some logging and reduced the overall interface. The Insert menu has been included to illustrate the added functionalities. An instructor could now choose, with the push of a button, to duplicate the background of a slide (an image inserted onto a slide). In later versions we would allow an instructor to duplicate the ink on a slide, the background image on a slide, or any combination of each.

also included some logging of application usage. In the beginning stages of application development we were primarily focused on building a powerful tablet-based presentation system. At the same time, we realized that capturing user interactions could facilitate software development and provide meaningful data on Tablet PC usage patterns.

Figure 5.4 illustrates e-Fuzion 2005 version 1 of the stand alone application. This release included a number of features: zooming (which provides for an expansive whiteboard canvas), scroll bars, drop down menus for quicker navigation of slides, and a screen capture tool. An interesting addition was the white-out tool which essentially implemented white liquid paper by adding white ink over any surface. The major modification of the user interface was the increase in the size of the icon buttons and the inclusion of toolbars that could be docked and undocked from the application window. The “Modules” menu contained a screen snapshot tool called cropper written by Brian Scott allowing for the direct importation of any screen content selected by the tool. The inclusion of scrollbars and zooming capabilities provided some flexibility for navigating the whiteboard canvas.

Having received feedback from users of the system we needed to make several improvements. For example, the icons were too small for the lecturer to manipulate with the pen stylus and the zoom in and zoom out facility was difficult to control. We also found that clicking the icons for functionalities such as duplicating page contents was much easier than trying to select the same functionalities from the text-based menu. The drop down style of the duplicate page button required an extra click and careful navigation and selection with the pen stylus (figure 5.5).

The e-Fuzion 2005 application integrated a network protocol for the potential sharing of whiteboard contents across users. (We will talk about the network-based applications in subsequent chapters of this thesis.) A login prompt (figure 5.6) provided a user with the option of creating a server, connecting to a server, or running as a stand-alone application.

The updated application interface is illustrated in figure 5.7. Note that the navigation tools (previous page and next page) were one means to move from one slide to the next. Additional options were available in the text pull-down menu and in the “Select Slide” drop down menu; however, as stated above, manipulating the pen stylus proved difficult for some users. It is easy to see in hindsight we required a user interface design that could maximize screen real-estate by not crowding the screen with lots of icons and unnecessary functionalities.

Figure 5.8 illustrates the updated interface integrating support for textual input, larger icons, and transparency. This is the first version where we integrated the Agilix Infinotes (<http://www.agilix.com>) scribble

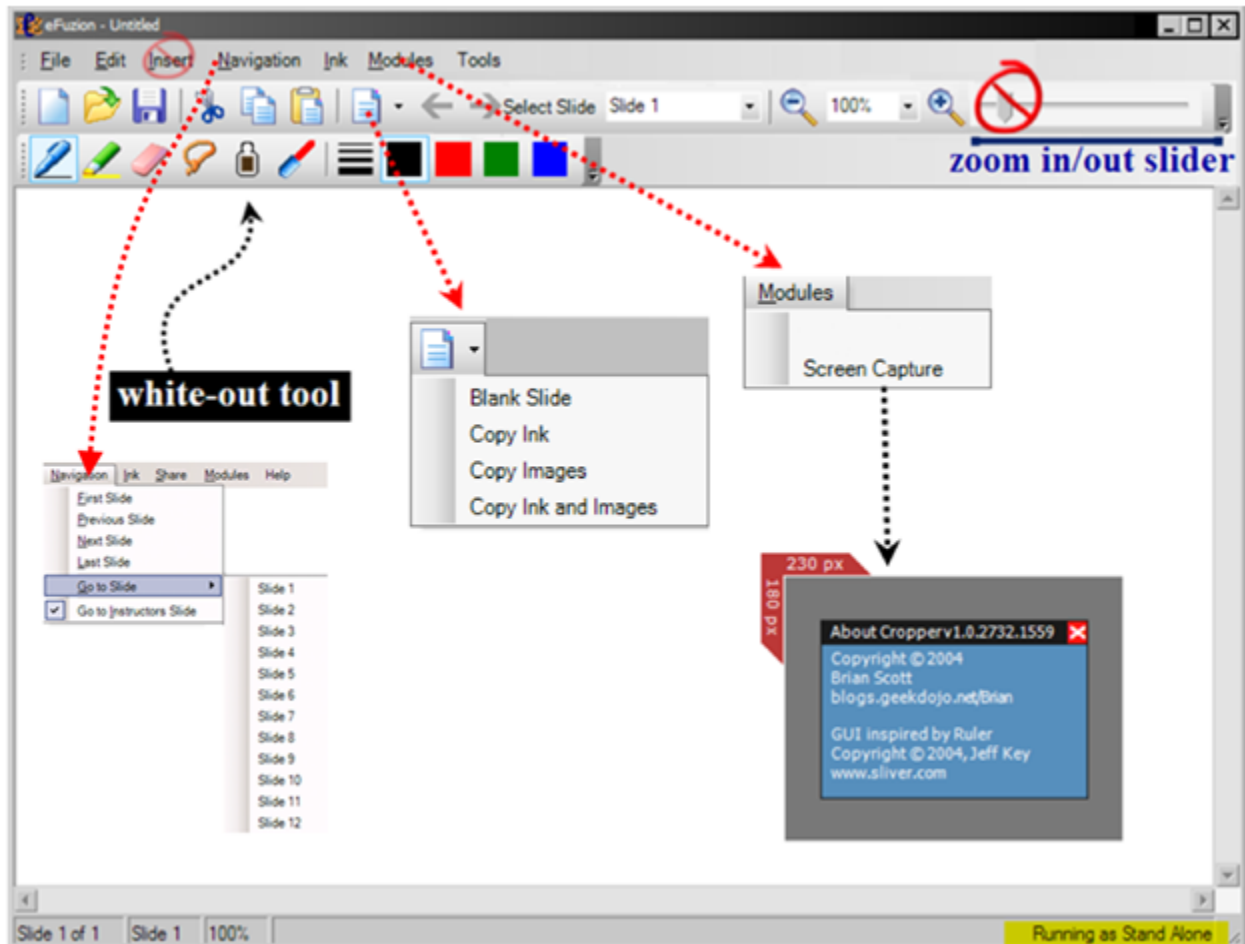


Figure 5.4: e-Fuzion 2005 version 1. The icons were too small for the lecturer to manipulate and the zoom in/out slider was not very useful. An interesting addition was the white-out tool which essentially implemented the white liquid paper by adding white ink over any surface. We also found that clicking the icons for duplicating page information and inserting their contents was better than trying to select the same functionalities in the text-based menu. Still, it would require too many clicks and careful navigation with the stylus. The Modules menu contained a screen snapshot tool allowing the direct importation of any content selected by the tool.

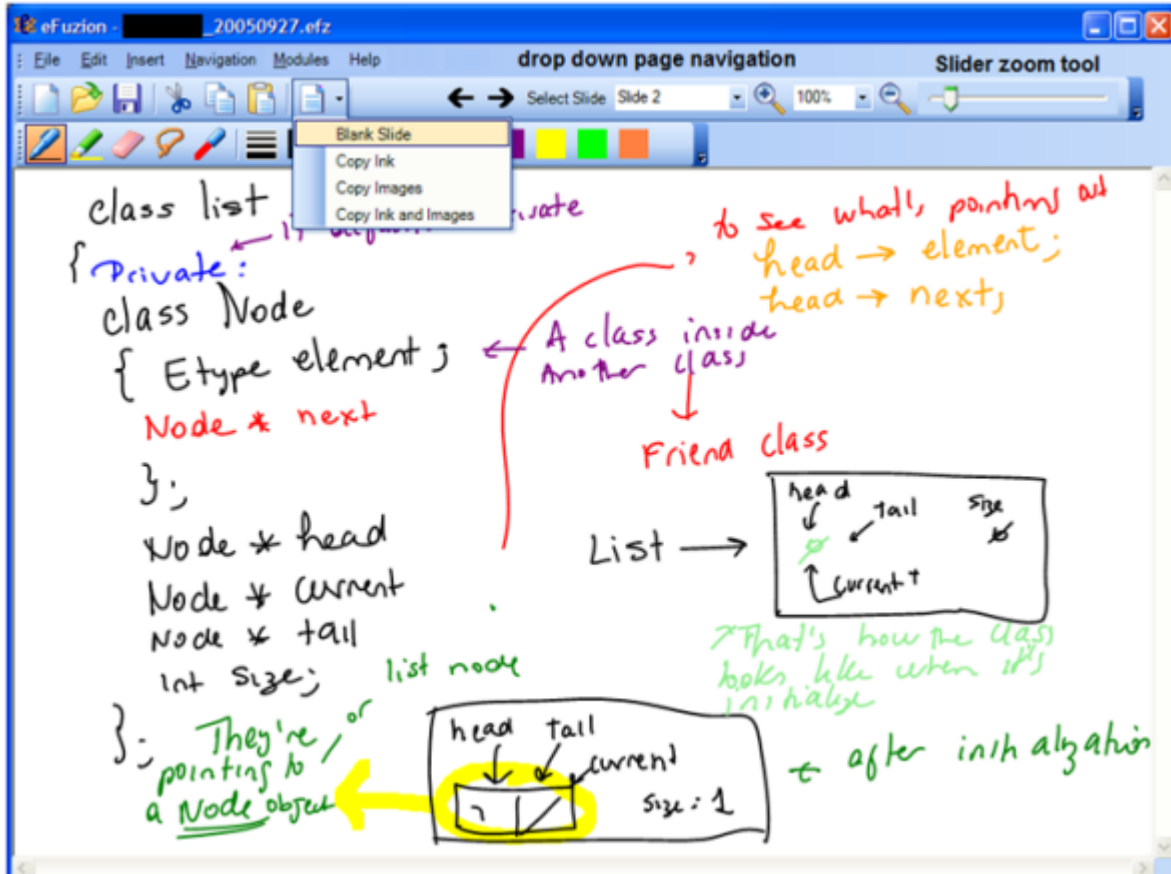


Figure 5.5: e-Fuzion 2005 version 2 stand alone application. The whiteboard contents include a snapshot of a student's notes on Linked Lists enrolled in CS400 in the fall 2005 semester. This release fixed a number of bugs found in the previous release. The additional colors were of use to a number of students, but overall the interface was cluttered with unnecessary buttons and menu options.

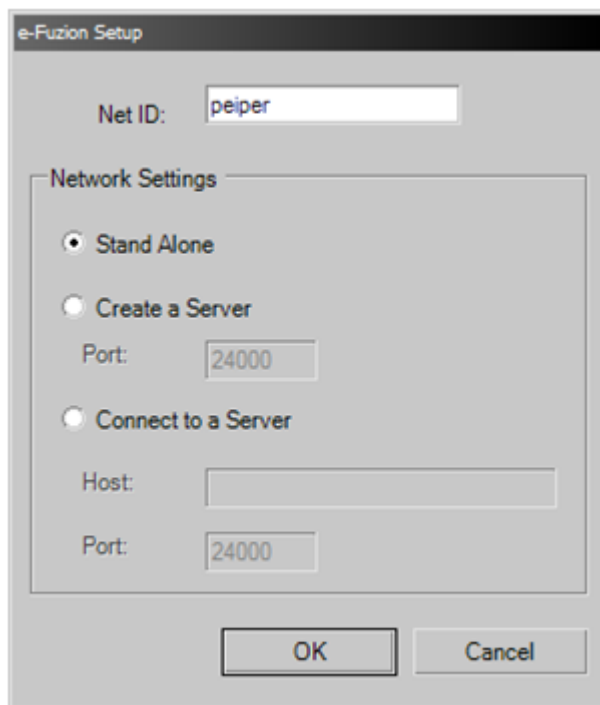


Figure 5.6: Login screen for e-Fuzion. In the instructor-only (stand alone mode) a net id is required for identification purposes when logging the user's tablet interactions. In chapter six, we will examine this version of the application where the Net ID is used for identification verification and for collaborative sharing of whiteboard contents between students and instructor.

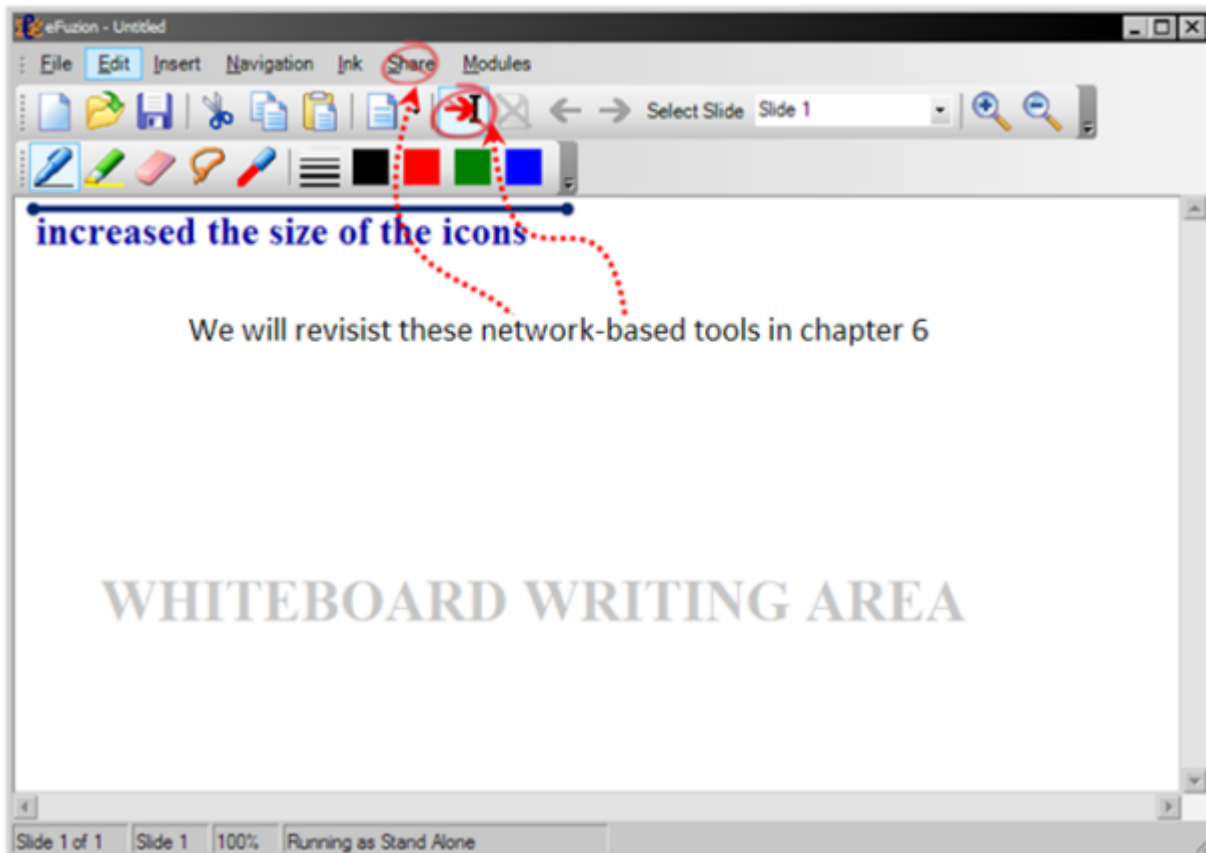


Figure 5.7: The modified e-Fuzion 2005 version 3 stand alone and network versions for sharing ink. The only modifications made to the system were the increase in the icon sizes for easier manipulation with the stylus and the removal of the zoom slide bar which was extremely difficult to use.

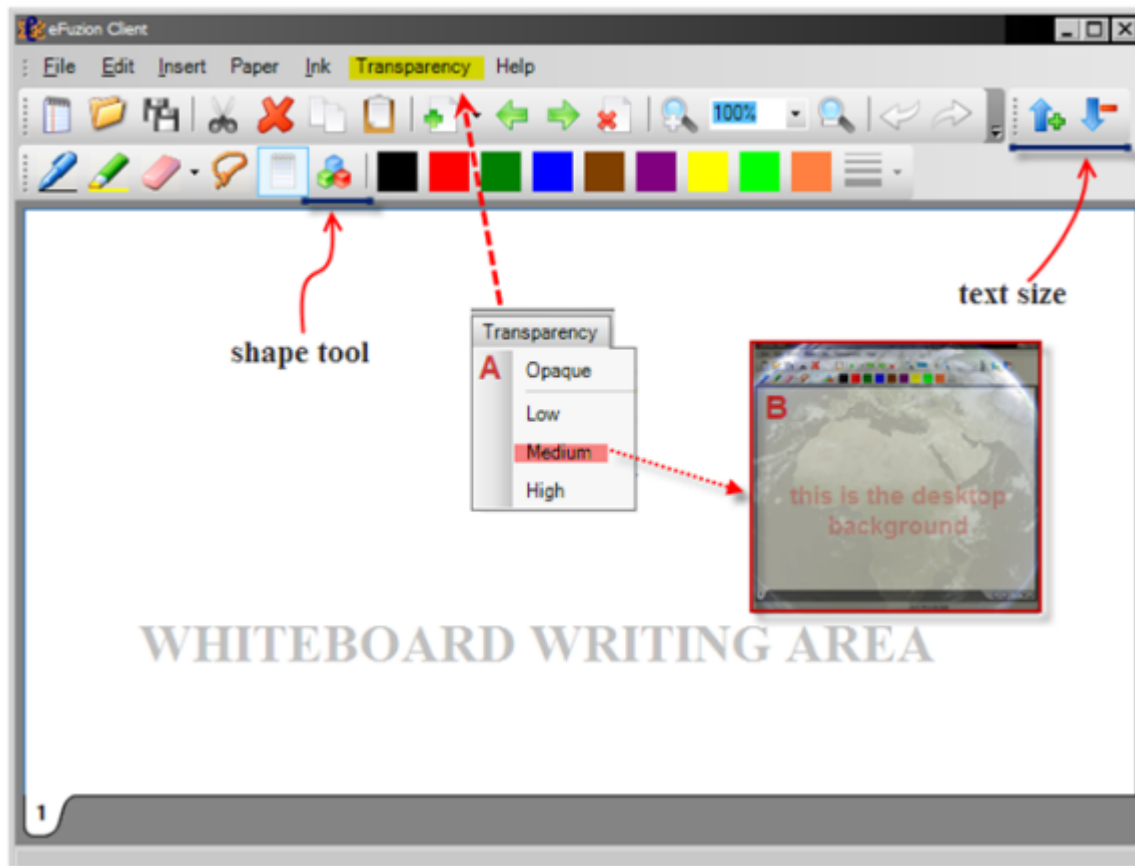


Figure 5.8: e-Fuzion Stand alone. This is the first version where we integrated the Agilix Infinotes scribble whiteboard into the system. The Agilix improved the performance of the whiteboard user experience by smoothing out a number of underlying functionalities in the Microsoft Inking API. Some other added features were rich textual tools, shape tools with an ink-based shape recognizer, and the ability to change the underlying background in the “Paper” menu (known as stationery). We added a “Transparency” menu option (A) so that an instructor could annotate “on top” of anything displayed on the screen (B).

whiteboard into the system. The Agilix¹ improved the performance of the whiteboard user experience by smoothing out a number of underlying functionalities in the Microsoft Inking API. Some other added features were rich textual tools, shape tools with an ink-based shape recognizer, and the ability to change the underlying background in the “Paper” menu (known as stationery). We added a “Transparency” menu option so that an instructor could annotate “on top” of anything displayed on the screen much like WriteOn [185].

e-Fuzion 2006 is illustrated in figure 5.9. The user interface, designed and programmed by Boris Capitanu, was a vast improvement over the previous interfaces. Aside from stripping the textual menu options to maximize the pen-input modality, the location of the tool bars was changed. An additional improvement later made would be the enlargement of the icon buttons. This made it much easier for an instructor to

¹To our knowledge, Agilix no longer supports the Infinotes product.

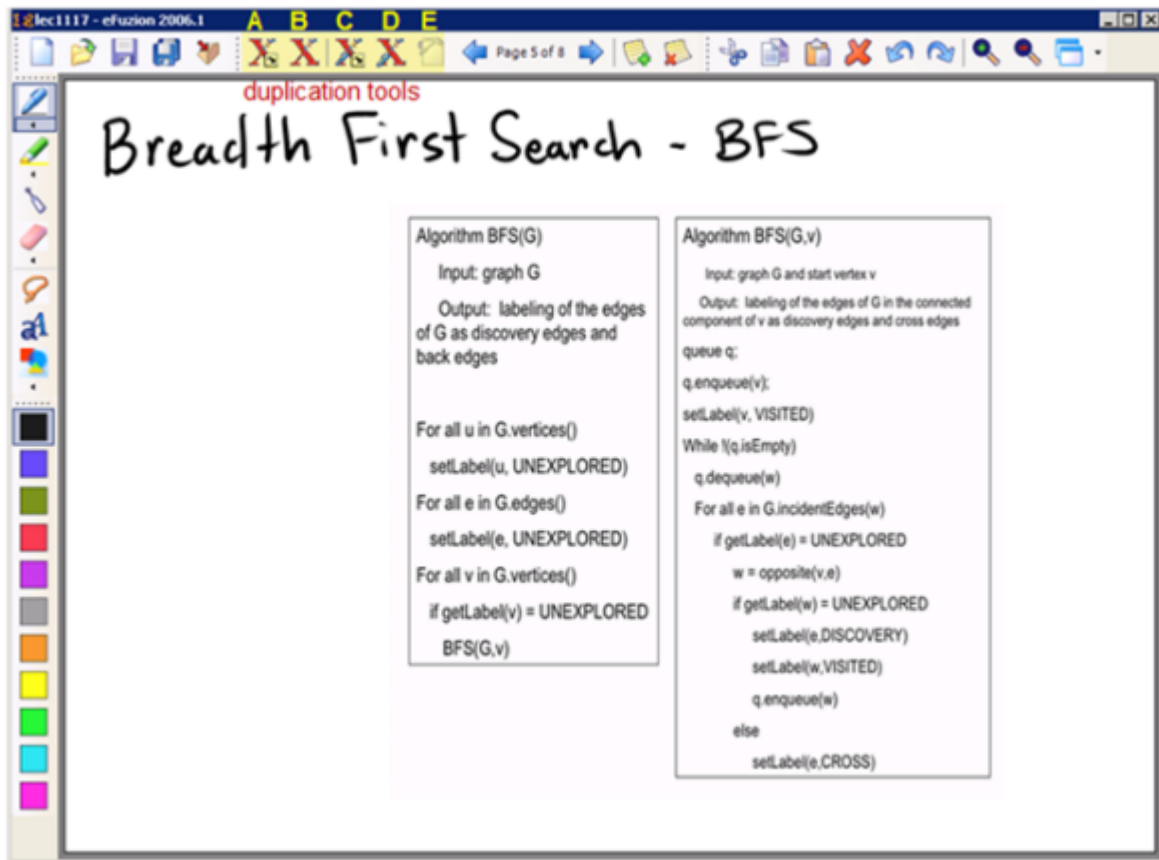


Figure 5.9

select a tool, by requiring less than perfect precision of the stylus. The technical problem we are referring to is the need to calibrate the digitizer on occasion, something the instructors were not really used to doing. Boris's artistic talent is also evident in the icons he created for the purposes of duplicating page content.

The final instructor version used for the Spring 2006 until the concluding Spring 2008 semester was written by Boris Capitanu of NuPaper® Software Inc. (figure 5.10). We were provided with a beta license to use in our classrooms.

5.3.1 New design considerations for the Tablet PC

The major difference between this version and the version described in chapter 4 was the utilization of the Microsoft® rich inking API. The active digitizer built into the Tablet PC was integrated into the Microsoft Ink application programming interface included in version 1.7 of the Tablet PC SDK. This SDK allowed developers to integrate the high sampling rate of the Wacom (<http://www.wacom.com>) active digitizers allowing for a higher resolution of pixel-to-screen ratio. The active digitizer provided visual feedback as the stylus hovered over the digitizer and it also increased the accuracy of the pen input decreasing the number of

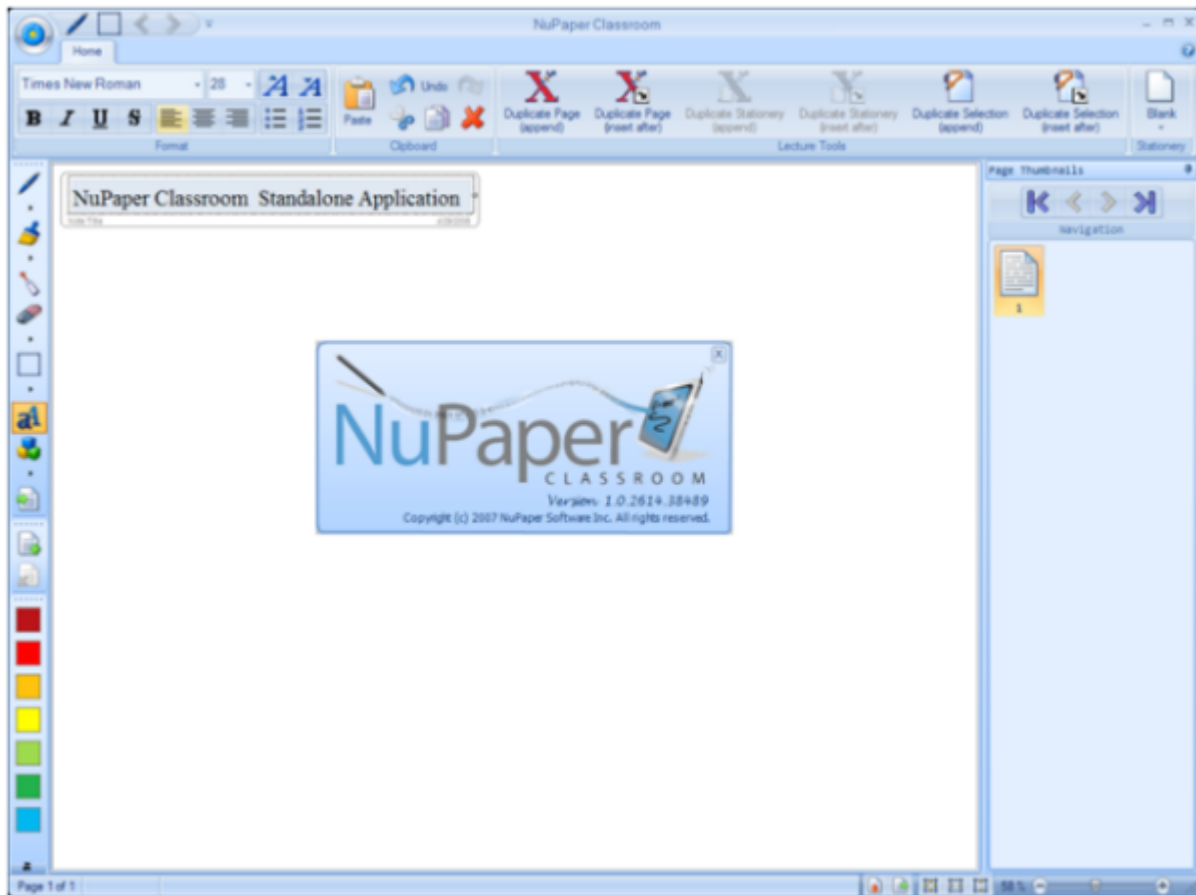


Figure 5.10: NuPaper Classroom® standalone presentation application.



Figure 5.11: The ink in this illustration was created in the NetMeeting® Whiteboard application. Note the degradation in quality when magnified.

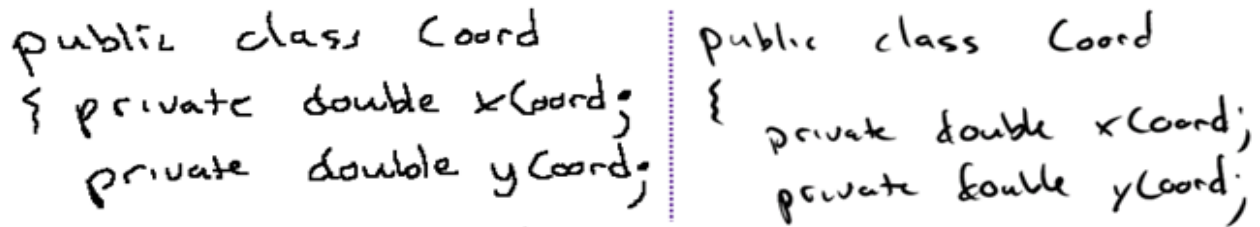


Figure 5.12: A comparison of the NetMeeting® Whiteboard application and the e-Fuzion application incorporating the use of the Microsoft® inking API. The automated reduction in the image size to fit into the above figure does not clearly illustrate the poor quality of the differences in legibility.

erasures. The lecture notes became more legible and therefore useful to students post-lecture (figure 5.11).

In this version of e-Fuzion we added a laser pointer tool to support the “electronic highlighting” of whiteboard contents. The big difference between the laser pointer and pen tool was that ink strokes created with the laser tool were always colored red and were not permanently displayed on the screen. As soon as the pen-stylus was lifted away from the digitizer any ink strokes created with the laser tool were removed giving the illusion of a laser. For more visually oriented students, this augmentation of the instructor’s gesticulations and sudden changes in voice inflection was most welcomed. The placement and choice of colors was also modified to allow for easy access and greater visibility. Except when using the highlighter tool to provide contrast, some of the lighter shades of orange, yellow and green were not as easy to discern when projected using the in-class projector. An example of using different colors is illustrated in figure 5.14.

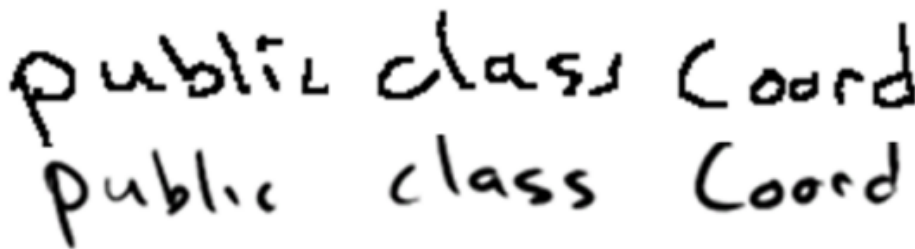


Figure 5.13: A magnified comparison of the NetMeeting® Whiteboard application and the e-Fuzion application incorporating the use of the Microsoft® inking API.

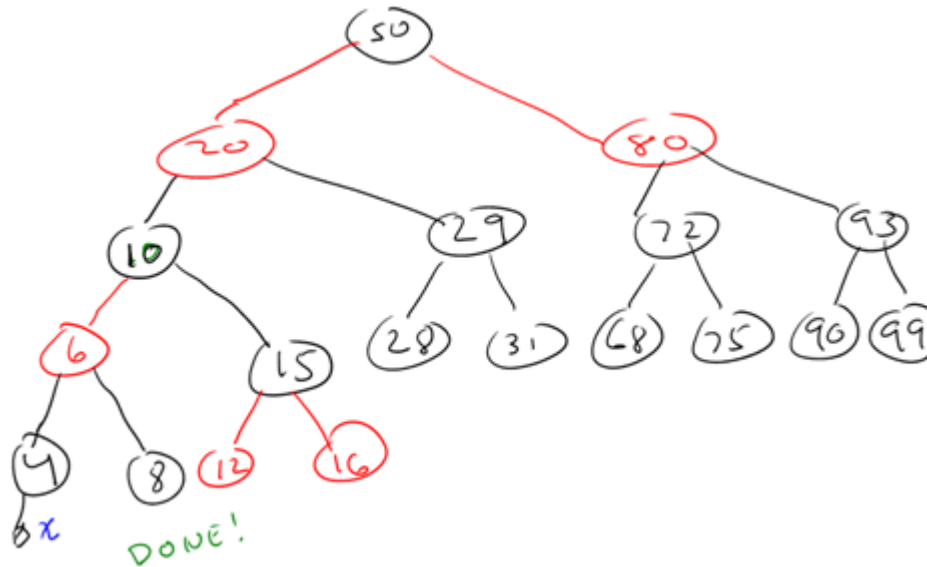


Figure 5.15: An excerpt slide of a CS225–Data Structures and Software Principles lecture on red-black balanced trees. This slide represents the final state of a rotation.

phases where an ink-drawn diagram progresses through several stages, each with a different meaning and purpose [8].

The WYSIWYG (What You See Is What You Get) principle warrants the recording of the teaching process. Students in the lecture hall felt more confident that the posted lecture notes would provide a meaningful context for review. While the addition of pages created in the lecture document increased the file size and the number of pages, students commented that reviewing the notes after the lecture was much easier.

An interesting dilemma results from the added whiteboard instructor tools. Ideally, slowing down an instructor’s interaction with the tablet could help provide students with more time to digest lecture material and provide more time for students trying to take notes.

One last problem with this model was that we had no way of understanding how the tool was being used without being physically in the room during a lecture. We needed a way of recording an instructor’s interaction with the tablet to facilitate future design of our system. It was through analysis of these logs (in subsequent versions of the software) that we realized precious time was spent creating new pages and copying and pasting lecture content. Regardless of our expectations, e-Fuzion seemed to be an improvement over the original NetMeeting Whiteboard application.

As a result we began experimenting with logging the instructor’s ink strokes in an attempt to investigate how ink was being used in the application. Aside from recording all the ink strokes created in lectures,

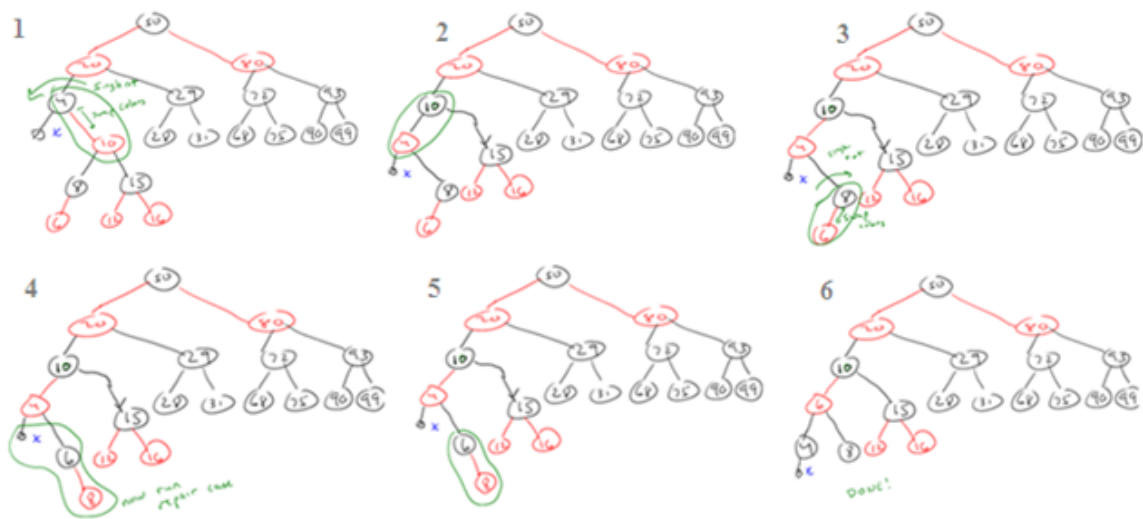


Figure 5.16: An excerpt of a CS225–Data Structures and Software Principles lecture illustrating the use of copy, insert new page, and paste ink objects to record the different states of a red-black tree rotation.



Figure 5.17: An excerpt of a CS125–Introduction to Computer Science lecture illustrating the use of copy, insert new page, and paste ink objects to record subsequent method calls

live observation provided information that aided the design process of our application: improvements of currently implemented functions and ideas for new functionalities that might be integrated into subsequent releases.

At the conclusion of the semester, after reviewing log files from the previous semester, it became apparent almost immediately that recording only ink strokes in our logs was insufficient if we were trying to gain insight into overall application usage. While classroom observations allowed us to accumulate ideas on the usability and technical problems that existed in our software, we believed that more complete log files could provide a more detailed account (such as the recommendation for “higher fidelity” by the e-Class (section 2.5) authors) of what transpired in lecture. For example, the creation of the laser pointer tool was a result of observing the instructor’s presentation. Nothing in the log files showed a need for such a tool. While the lectures were recorded for later observation and coding, the information saved in our log files was insufficient. By recording additional user interactions we could have attained a great deal more information. For example if we had had a better understanding of how often the instructor used the copy and paste tools, we would have saved valuable teaching time by providing a tool for duplicating and inserting previously written content. This, among other things, was integrated into subsequent releases of the application.

5.4 Classroom Survey Evaluation Results

We have three classes (taken over the course of two semesters) of survey data where we asked students to fill out questionnaires regarding their use of instructor-created whiteboard notes and their perception of how well the technology enhanced (if at all) the learning environment. In spring 2007 we surveyed two classes at the conclusion of the semester: CS125–Introduction to Computer Science and CS225–Data Structures and Software Principles. In the fall 2007 semester we surveyed the CS225 course. Surveys were completed by students prior to handing in their final examinations. In total there were 333 surveys completed. The student questionnaire is available in appendix A.1 and all the results may be seen in appendix A.1. In this section we summarize the overall results and provide illustrations of the interesting measurements.

The first question of our survey revealed that 30% of the students did not “regularly attend lecture” (figure 5.18). One motivation for asking this question was to get an idea of how many students regularly skipped lecture. We also wanted to try to determine how many surveys were submitted by people who almost never attended class. The integration of the e-Fuzion technology seems to support a lack of concern for missing lecture (figure 5.19). Of course, it does not provide definitive evidence, but illustrates that students were confident that post-lecture documents would be posted to the course website, and that the content of

Regularly Attend Lecture?

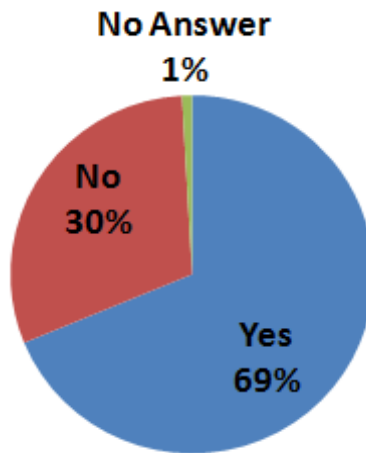


Figure 5.18: 30% of the students enrolled in the spring 2007 CS125 and CS225, and fall 2007 CS225 class regularly skipped lecture.

these documents would be useful. (See appendix D.1 for three examples of posted lecture files; a completed instructor-annotated file, a skeletal form of notes posted before the class lecture, and a subsequently version of the latter annotated by the instructor.)

Figure 5.20 reports that 35% of the students printed the instructor’s skeletal slides before lecture. This was easily confirmed by a cursory viewing of the course site webpage access logs. Although we cannot deduce from this statistic alone whether or not having the skeletal form of the instructor’s presentation facilitated student interaction, 67% of the students agreed that the classroom lecture was more engaging as a result of the technology (question A.2) and 46% believed that the availability of notes post-lecture made it “easier” to pay attention (question A.3). An overwhelming number of students believed that technology was used effectively (73%, question A.4), and that the course was well suited for the software (78% question A.5). All things being equal, 70% of the students said they preferred to take a class that uses e-Fuzion technology over the same class that does not (question A.11).

When we asked whether or not a student’s note taking practice had changed, we were surprised to learn that 45% of the students reported that they changed their note taking style in some way (figure 5.21). Unfortunately we do not have the data to support or suggest why or how the students’ note-taking practices had changed.

Overall, the perception of the students enrolled in the three courses over the span of three semesters was extremely positive. As a result, the instructor has become dependent on the e-Fuzion application to support her teaching.

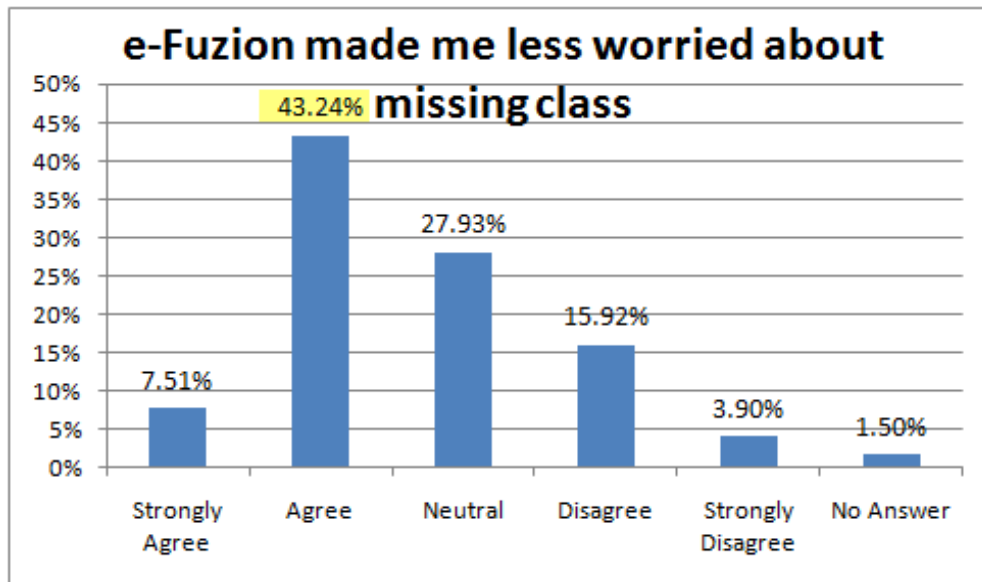


Figure 5.19: 50% of the students believed that e-Fuzion made them “less worried” about skipping class. One reason could be that close to 70% of the students “trusted” that the notes would be posted to the course website after class (question A.14) and that 21% of the students relied on accessing the notes as a result of missing class (question A.10).

Would you print slides before class?

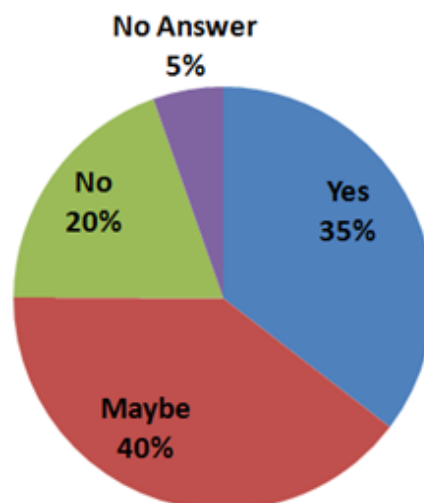


Figure 5.20: 35% of the students printed the lecture slides before class.

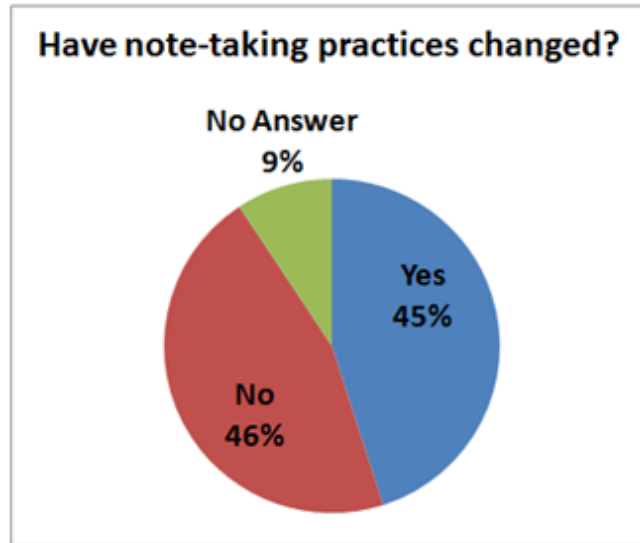


Figure 5.21: 45% of the students reported that their note-taking practices had changed.

5.5 Data Analysis

Affordances of the pen-based technology with the active digitizer² allow us to collect interactions with the Tablet PC that we used to query and measure a number of metrics related to an instructor's use of the technology in the classroom. We are assuming an application like e-Fuzion which supports taking notes on a whiteboard with the pen-input device. We required the use of either a slate (a tablet without a keyboard) or convertible tablet (see section 3.3.2), which provides a keyboard, although the results presented in this chapter omit the use of the keyboard as an input device.

Pages of a document created in our presentation software (e-Fuzion) running on a Tablet PC contain low-level data that we capture for later analysis. The finest granularity of information is at the ink stroke level. Each ink stroke contains the following data: pressure information (how hard the instructor is pressing down on the digitizer pen), X and Y coordinate location of individual points making up the stroke, time data of the stroke creation (when the stroke was started), the duration of the stroke (how long in milliseconds the pen digitizer captured information), and the color of the ink stroke. Microsoft defines an ink stroke as:

a set of properties and point data that the digitizer captures that represent the coordinates and properties of a known ink mark. It is the set of data that is captured in a single pen down, up, or move sequence (source: <http://msdn.microsoft.com/en-us/library/ms827842.aspx>).

Each page in our document typically contains a collection of ink strokes. Therefore, each page can provide

²See <http://msdn.microsoft.com/en-us/library/ms818317.aspx> for a description of accessing and manipulating pen input and <http://msdn.microsoft.com/en-us/library/aa813347.aspx> for a description of the *realtimestylus* and ways to filter information derived from the Tablet PC digitizer.

for summary information by aggregating and analyzing the collection of strokes including how much ink was created (in some human readable form such as total inches), the number of strokes, the average, speed and pressure information of the stroke, and a measurement of change such as the number of strokes deleted. There are a number of different tool sets available in the note-taking application such as writing tools which include a pen tool, a highlighter tool, and a laser pointer tool. Editing tools include an erasure tool and the selection/lasso tool which allows for the modification of an ink stroke or collection of ink strokes. These modifications may include changing the location, size (including thickness) and color attributes of the selected data.

More common tools typical of whiteboard applications of the past include the insertion of shape objects, pictures, web links, textual input, and even blank space. The copy, cut, paste, and delete commands exist for both ink strokes and for other objects inserted on the whiteboard such as pictures or shape objects. Undo and redo allow for recovery from mistakes, and a customizable color tool bar provides for a palette of colors to choose from. The navigation tools allow for the traversal of pages within a document. In earlier versions of our presentation system instructors were forced to navigate through their documents in a linear fashion or skip to the first or last slides respectively. In later versions we provided thumbnails to provide instant access to all slides in the document. Document tools such as the adding and removing of pages were enhanced to include tools such as the duplicate page tools which create a new page by copying specific contents (copy all objects, copy only ink strokes, duplicate background or stationery, and selection-based copy) and either inserting or appending the copied page to the document. View tools included zooming in and out, fit-to-page, fit-to-height, and fit-to-width.

The format of the log files changed slightly with each update to the e-Fuzion whiteboard application. We provide an excerpt of a log file from the spring 2005 semester highlighting the data representation of an ink stroke (figure 5.22). The packet description includes pressure information and timer tick information. The pressure information is measured on a scale from 0 to 254 and is automatically associated with the n^{th} point in the timer tick list. For example, the highlighted normal pressure points 101, 118, 129, 137, 144, and 149 represent the pressure information as recorded by the digitizer at times (in milliseconds) 21, 31, 41, 61, 71 and 91 respectively. Note that the timer tick information sometimes reports duplicate values and is usually (depending on the quality of the digitizer technology) sampled every 10 milliseconds.

In this section we present examples of data analysis based on the classes of Jason Zych, a very popular and effective instructor (no longer with the department). Jason received his B.S. and M.C.S. in Computer Science at the University of Illinois at Urbana-Champaign. As a graduate student in the department Jason dedicated much of his time to teaching. As the head Teaching Assistant to a mix of Professors assigned to

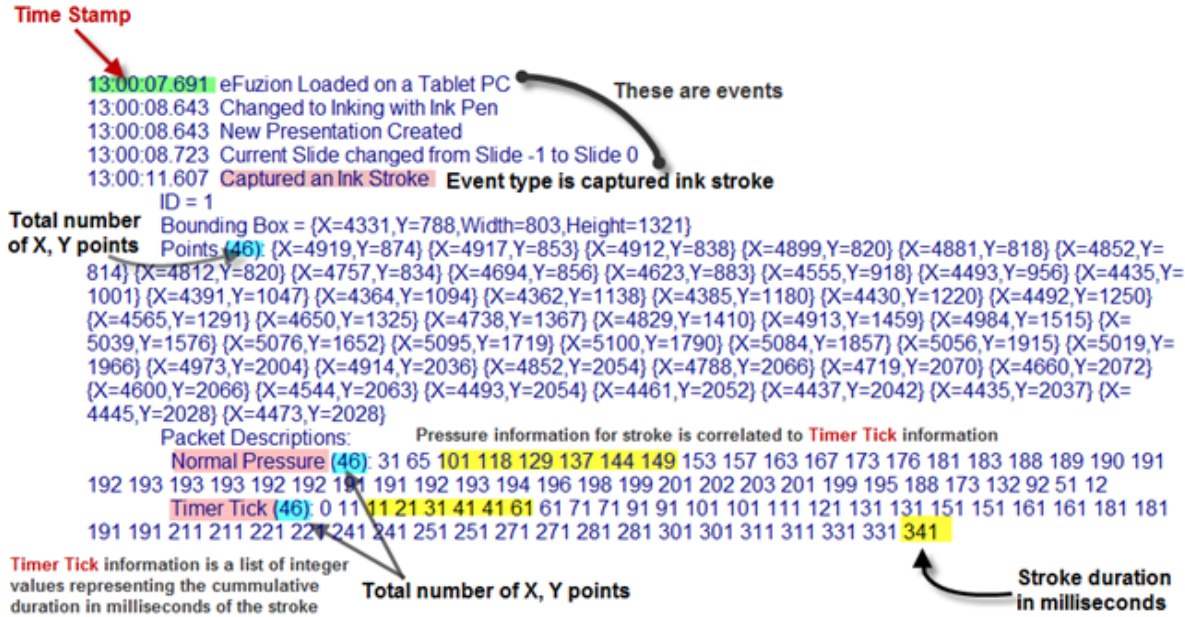


Figure 5.22: Example log file used to capture application usage in the spring 2005 semester

Table 5.1: Spring 2005 teaching schedule

Course	Section	Time	Day	Enrollment
CS225	AL1	12:00pm-12:50pm	MWF	117
CS125	AL1	1:00pm-1:50pm	MWF	182
CS125	AL2	2:00pm-2:50pm	MWF	134
CS225	AL2	3:00pm-3:50pm	MWF	89

teach CS225 (our foundations course in data structures and software principles) Jason became extremely popular and was regularly placed on the “List of Teachers Ranked as Excellent” by their students. Jason was hired as a Visiting Lecturer to teach the summer session CS125 and CS225 courses in the summer of 1998. Soon after, Jason became the first permanent “Lecturer” hire in the department and was assigned the responsibility of teaching two of the core courses in the undergraduate curriculum. Table 5.1 presents the spring 2005 teaching schedule of Jason Zych.

As a point of introduction we present an example of one lecture summary (figure 5.23) for one of the four classes taught on February 9, 2005. We have summaries for all forty lectures for each of the four classes (160 in all)³. Each summary contains a mix of average aggregate information and actual tool usage represented by raw count and duration in minutes. Part A of the figure summarizes average aggregate information for each page (column 1), giving number of strokes (column 2), average ink stroke length (column 3), average ink stroke pressure and average ink stroke speed. Part B of the figure provides raw counts of tool usage

³ The use of the micro charts (http://www.bonavistasystems.com/Products_SparkLiner_Overview.html) made it a lot easier to present twelve lectures at a time for variation, low and high values.

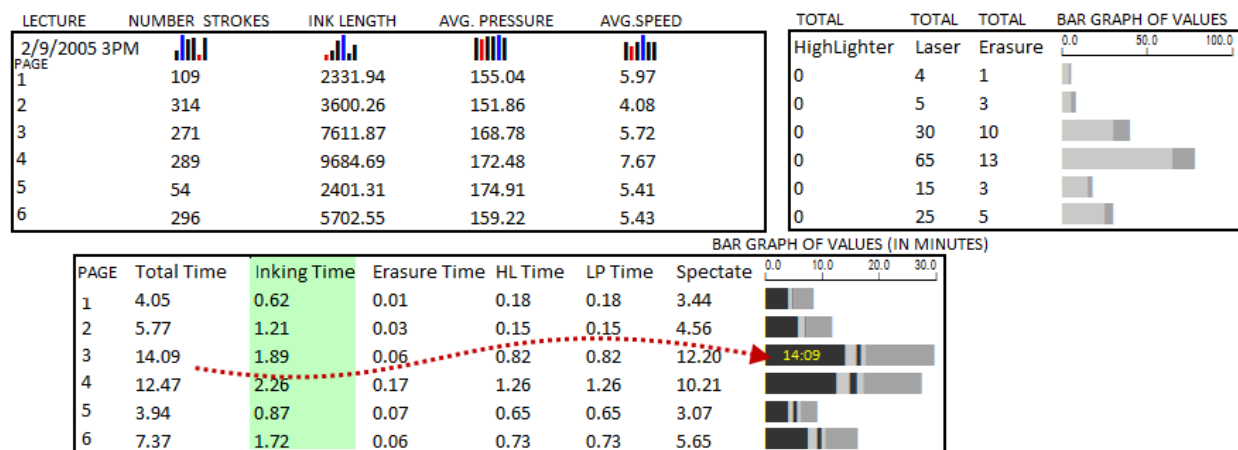


Figure 5.23: Lecture summarization of whiteboard usage

including the number of strokes created with the highlighter, laser pointer, and erasure tool for each of the six pages of the document. Notice the number of laser pointer and erasure usages in pages two and three, (30, 10) and (65, 13) respectively. In part C of the figure, time measurements (in minutes) for tool usage are summarized. Note that the time spent “inking” (the highlighted column’s values) is much less then the “spectate” time (time where the page of a document is displayed but no tools are used) and the “total time.” The corresponding six pages of the document summarized in figure 5.23 are illustrated below.

We organize the data analysis according to a number of different correlations: given any attribute recorded during tablet interaction by an instructor we can compare the raw data at different levels of granularity. For example, we might choose to look at speed of writing versus time within a lecture, within a semester, within a single day (Jason Zych taught four classes straight from 12:00PM–4:00PM Monday, Wednesday, Friday), or within a single page. We can make comparisons between the same subjects (two different semesters of CS225), different subjects (CS125 vs. CS225), and between the same instructors (two semesters of CS225 taught by Jason Zych), and different instructors (CS225 taught by Cinda Heeren vs. Jason Zych in tables 5.2 and 5.8).

In looking at the speed of writing we might choose to investigate whether an instructor changes the way he or she writes. For example, is the instructor writing faster, slower, longer, or shorter in one lecture than another? At what point in the lecture does this change occur? Is it consistent across instructors? Do we see the same behavior in the same courses and or different courses? Several categories of questions seemed interesting to us.

We realize that there are a myriad of research questions that we do not address. In this chapter we only wish to illustrate the potential for researchers to address and ask questions they could not before the advent

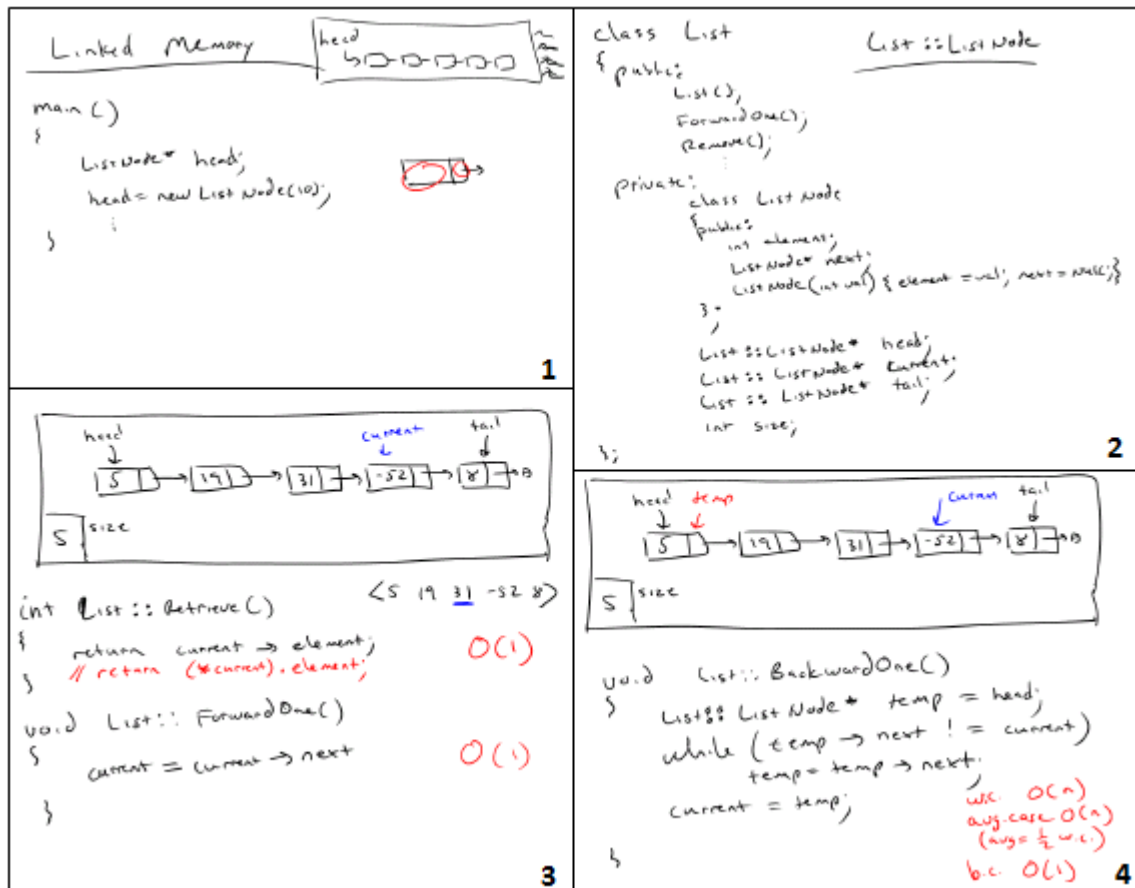


Figure 5.24

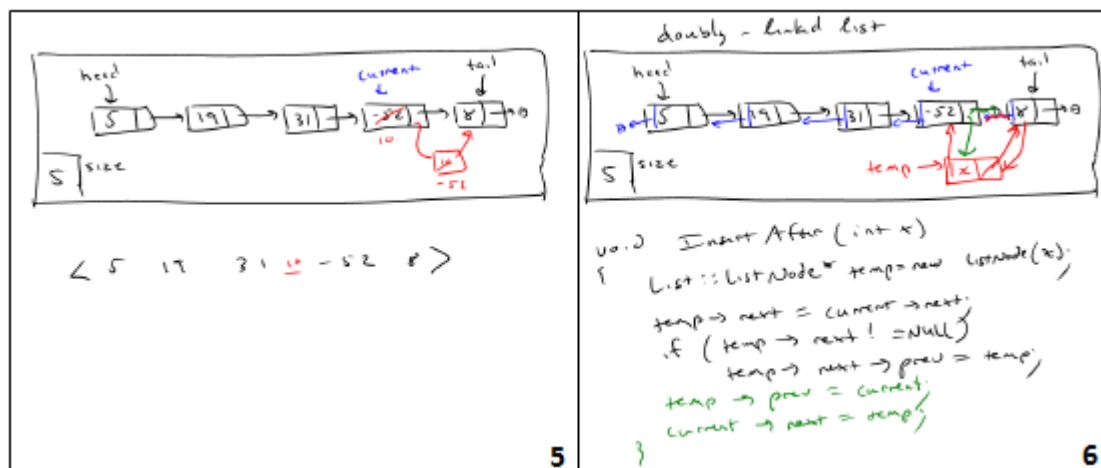


Figure 5.25

of the Tablet PC and rich ink technology.

5.5.1 Basic summary information

How much time was spent on each page in the lecture presentation? How much time was spent writing vs. not writing on each page? How fast did the instructor write, and did his or her speed vary during the class; if so, in any predictable way, or just randomly? How often did the instructor change slides, change pen colors, erase strokes, and use other tools available on the Tablet PC application? The interest in this data arises largely from the fact that it has not to date been available anywhere else. This is the kind of “baseline” data that is easy to gather when the instructor writes on a computer, but virtually impossible otherwise. It also seems reasonable that researchers interested in pedagogy would be interested in these data as a means to measure, assess, and evaluate how teachers teach. The data is collected unobtrusively [193] and as we will show in subsequent sections, has the potential to address research questions not previously possible.

Figure 5.26 presents a reduced summary of ink stroke data in the form of four attributes of ink strokes: quantity (number of inks strokes) and the length⁴, pressure, and speed value for all the strokes (column 1) on a given page. The length, pressure, and speed columns are highlighted green to emphasize that their respective values are averages. According to the summary information the CS225 12PM lecture on January 31, 2005 contained ten pages (each page represents a row in the table) and the column attributes represent an aggregate of stroke attributes at the page level. The mini-charts are graphical representations of their quantitative counterparts. Figure 5.27 contains thumbnails of the document summarized in figure 5.26. The average ink length and speed values for pages two and three are highlighted in yellow (in figure 5.26 and bordered in red in figure 5.27) to call attention to their respective values.

Whether a correlation can be made between average ink length and average speed (for example, does a larger ink stroke length imply a faster drawing speed?) has yet to be determined and requires analysis. It is interesting to note that we have seen a trend in a number of lectures where the last page of the instructor’s whiteboard document contains the maximum values for ink stroke speed and pressure. As an example, page 10 of the lecture document illustrated in figure 5.27 is bordered in blue to underscore the *largest* values for both the average pressure and average speed attributes, 184.08 and 15.31, respectively. While these observations might seem superficial and inconclusive, our hope is that they illustrate how a lecture summary of this kind can provide for additional information not easily available without the use of the Tablet PC⁵.

⁴The values for stroke length presented in this chapter are in inches. The original values were recorded in “ink space” which is a virtual coordinate space to which the coordinates of the Tablet PC are mapped. This space is fixed to a HIMETRIC coordinate system where 1 HIMETRIC unit = 0.01mm. See <http://msdn.microsoft.com/en-us/library/microsoft.ink.ink.aspx> for additional information

⁵We realize it is possible to capture characteristics of instructor interaction in a traditional classroom setting. However, the cost of the equipment required to capture, digitize, and code this data (not to mention the obtrusive nature of the “observation”





Lecture	Strokes	Ink Length	Avg. Pressure	Avg. Speed
1/31/2005 12PM				
1 PAGES	47	3424.13	183.46	8.42
2	203	22067.87	174.93	10.70
3	223	17441.38	173.58	8.93
4	73	1119.76	177.64	7.37
5	121	2027.04	169.01	5.09
6	224	6546.87	174.68	7.37
7	86	1037.65	174.94	6.28
8	68	1571.36	182.70	9.04
9	150	2008.73	176.34	6.43
10	73	7071.16	184.08	15.31

Figure 5.26: Lecture summary information

The Big Three

```

class Coord
{
public:
    Coord();
    Coord(double x, double y);
    Coord(const Coord&);
    ~Coord();
}

```

Page 1 of 10

Coord() ~ Coord()

```

{
    delete label;
}

```

default:
Coord() ~ Coord()

if empty

Page 2 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 3 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 4 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 5 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 6 of 10

operator overloading

↳ function overloading ... but
with operators!

```

+ (a, b)    operator+(a, b)
a + (b)    a.operator+(b)
operator[] (int i)
a[i]       a.operator[](i)
a = b      a.operator=(b)

```

Page 7 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 8 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 9 of 10

Coord() ~ Coord()

String + Label

Constructors for
primitives
types (int, double,
pointer) do
nothing!

Page 10 of 10

Figure 5.27: CS225 lecture six whiteboard document

Table 5.2: Comparison of spring 2005 total lecture attributes

	Pages	Total number of strokes			Total time in minutes		
		Ink	Laser	Deleted	Inking	Laser	Erasing
CS225							
12PM	451	48,506	3,678	769	274.78	75.71	9.95
3PM	586	54,033	5,073	919	331.47	109.52	10.74
CS125							
1PM	487	51,075	3,489	684	270.51	69.86	8.28
2PM	524	46,066	4,294	614	276.65	93.24	8.24

Table 5.3: Comparison of spring 2005 average lecture attributes

		Average number of strokes			Average duration in minutes		
	Pages	Ink	Laser	Deleted	Inking	Laser	Erasing
CS225							
12PM	12	1,373	97	21	7.20	2.00	0.30
3PM	14	1,375	118	21	7.71	2.55	0.25
CS125							
1PM	12	1,399	90	18	6.94	1.82	0.21
2PM	14	1,325	113	16	7.28	2.45	0.22

Table 5.2 contains summary information for all four classes taught by Jason Zych in the spring 2005 semester. The aggregate values presented in the table were derived from 39 of the 40 lecture log files recorded for the duration of the semester. The data presented in table 5.2 presents an interesting perspective on how Jason interacted with the whiteboard application in the process of instruction.

In subsequent sections we provide a more in depth overview of Jason’s teaching based on an analysis of 25 of the 40 log files recorded for each class during the spring 2005 semester⁶ and include average aggregate information for all four classes over the course of the semester.

Table 5.3 presents the average number of pages, ink strokes, and laser strokes created, the number of ink strokes deleted, and the amount of time (in minutes) spent. Table 5.4 presents the total number of pages, ink strokes, and laser strokes created, and the number of ink strokes deleted. Table 5.5 presents the total time spent (in minutes) using the ink, laser, and erasure tools. In table 5.6 we compare the length and speed of the ink and laser pointer strokes and table 5.7 presents the total length of ink expended in more “human readable” metrics.

In the following sections, we provide examples of the attributes summarized in the tables above. First we present example of total tool usage such as the total number of strokes created with the laser pointer tool, the total number if ink strokes created, and the total number of ink strokes deleted. We next present examples

device) might outweigh the benefit.

⁶The rationale for including a subset of all the log files in these analyses was to be as consistent as possible in our measurements. Some of the log files were not available due to circumstances beyond our control and as a result we chose to use the maximum number of complete log files for the *same* dates.

Table 5.4: Comparison of spring 2005 lecture stroke attributes

Class	Pages			Ink strokes			Laser strokes			Deleted strokes		
	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>total</i>	<i>avg</i>	<i>stdv</i>
12PM	271	10.84	2.57	33233	1379	285.11	2421	120	58.38	543	0.72	1.97
3PM	267	10.68	2.67	33105	1359	254.30	3177	152	69.01	558	0.16	0.36
1PM	315	12.60	2.68	37155	1561	353.55	2645	148	73.29	486	0.44	0.64
2PM	296	11.84	2.69	33292	1404	337.00	3307	179	88.87	426	0.68	2.36

Table 5.5: Comparison of spring 2005 lecture time attributes in minutes

Class	Inking time			Laser time			Erase time		
	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>total</i>	<i>avg</i>	<i>stdv</i>
12PM	133.15	5.48	0.91	51.02	2.79	1.42	7.51	0.37	0.22
3PM	135.21	5.56	0.86	72.13	3.56	1.55	5.92	0.31	0.15
1PM	140.90	5.87	1.06	54.34	3.23	1.66	5.80	0.27	0.16
2PM	128.81	5.45	1.24	72.21	4.19	2.15	5.10	0.26	0.14

Table 5.6: Comparison of spring 2005 stroke *length* (in inches) and *speed* attributes

Class	Ink length			Laser length			Ink speed		Laser speed	
	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>total</i>	<i>avg</i>	<i>stdv</i>	<i>avg</i>	<i>stdv</i>	<i>avg</i>	<i>stdv</i>
12PM	18202.36	0.55	0.91	10381.21	4.29	1.93	5.09	0.69	11.41	3.45
3PM	18189.18	0.55	0.86	13636.87	4.29	2.07	5.16	0.58	11.54	3.11
1PM	19095.41	0.51	0.11	13213.71	4.99	1.89	5.10	0.87	13.88	5.03
2PM	17338.45	0.52	1.24	16152.05	4.88	2.23	5.09	0.74	13.66	4.66

Table 5.7: Unit conversion comparisons of *ink* and *laser* length

Class	Ink length				Laser length			
	<i>inches</i>	<i>cm</i>	<i>yards</i>	<i>miles</i>	<i>inches</i>	<i>cm</i>	<i>yards</i>	<i>miles</i>
12PM	18202.36	46234.0	505.6	0.29	10381.21	26368.3	288.3	0.16
3PM	18189.18	46200.5	505.2	0.29	13636.87	34637.6	378.8	0.21
1PM	19095.41	48502.3	530.4	0.30	13213.71	33562.8	367.0	0.21
2PM	17338.45	44039.7	481.6	0.27	16152.05	41026.2	448.6	0.25

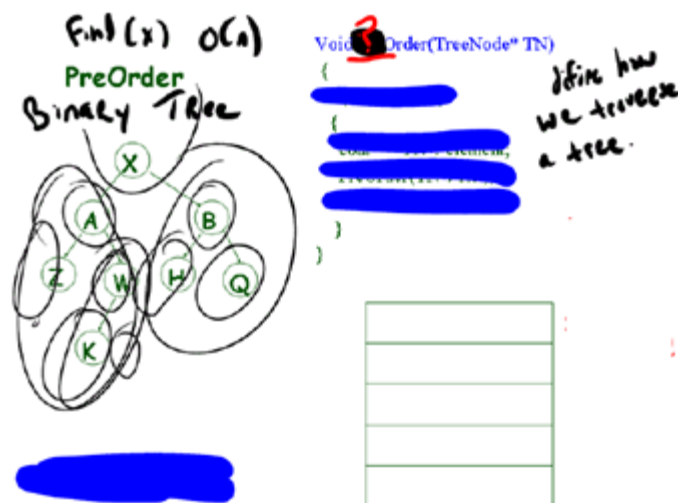


Figure 5.28: An example of too many “inked” annotations

of total time attributes which refers to the duration in minutes spent using the tools of the whiteboard application. These include inking time, laser time, and erasure time, which represent the amount of time an instructor spent creating ink strokes, using the laser pointer tool, and deleting ink strokes with the erasure tool, respectively. Lastly, we present the raw values extracted from the ink strokes recorded in the system. Recall that every ink stroke in the system contains a value representing its *length*, *speed*, and its *pressure*.

Total laser pointer

The laser pointer tool was a mechanism we built into the whiteboard application to simulate the laser pointers used to emphasize content on a projected screen. Using ink strokes to annotate portions of the screen for the sole purpose of placing special emphasis (such as circling an excerpt of code on the screen that an instructor is currently talking about) is more disruptive then helpful. Multiple annotations can quickly clutter the whiteboard space, hindering readability (figure 5.28).

Figure 5.29⁷ shows a comparison of the average number of laser tool strokes used per lecture in the 12PM CS225 and 1PM CS125 classes. CS125 has more laser pointer usage at the end of the semester than in the beginning. CS225, on the other hand, is more or less evenly distributed throughout the semester. The whiteboard page quickly accumulates useless ink strokes that need to be deleted. Using the Tablet PC we have the ability to record the laser pointer strokes just like we do ink strokes. In fact, the laser pointer is implemented by drawing red-colored ink strokes on the screen and immediately removing them from the canvas. It is also possible to customize the amount of time and the manner by which the laser pointer is

⁷Please note that the red line displayed in this and subsequent figures represents a two period moving average trendline and was added by us for emphasis.

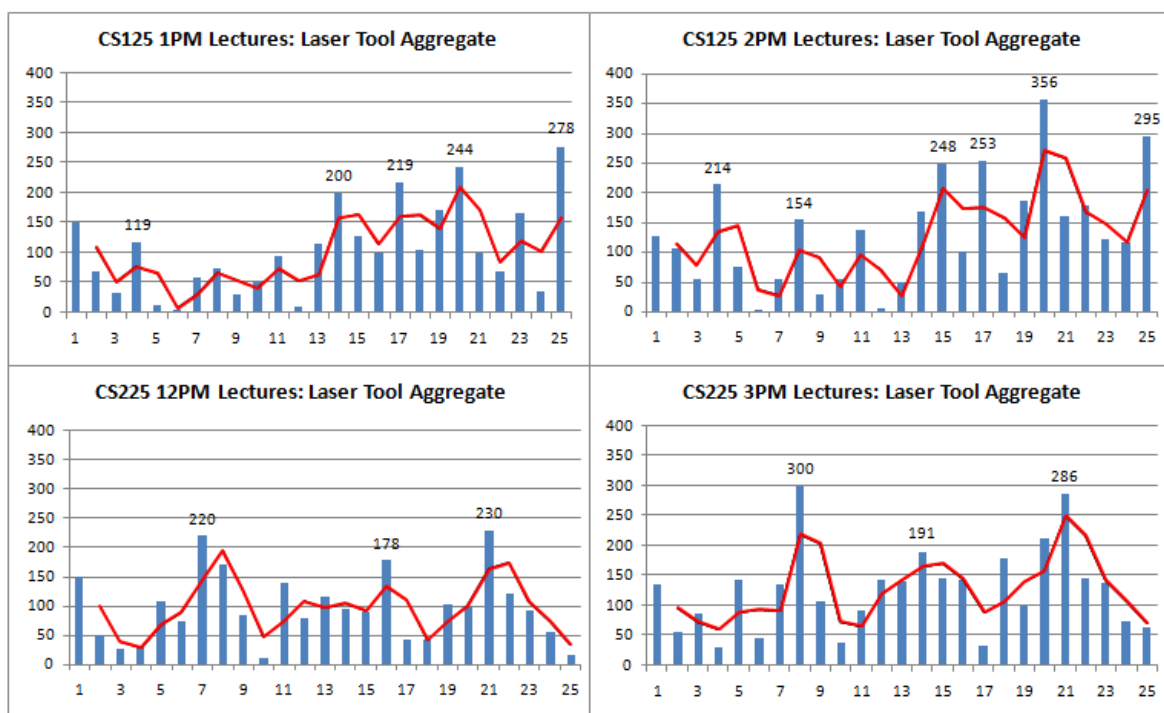


Figure 5.29: Average number of laser tool strokes per lecture for the spring 2005 CS225 and CS125 lectures.

displayed (i.e. three second pause and blinking). The novelty in recording the interaction with the tablet is that students can review the instructor-content post lecture. In the traditional classroom, you would not have a way to know what the instructor emphasized unless you were watching a video of the lecture (such as eClass presented in section 2.5) or if the document was marked up. The fact that the instructor used the laser pointer tool in class is a big hint to the students that the instructor was placing emphasis on specific regions of the whiteboard.

Total ink strokes

As another example consider the number of ink strokes (figure 5.30) which refers to the raw number of strokes created on average, for each page of the instructor's document. If an instructor was teaching two sections of the same lecture (as was done by Jason Zych) we would expect the number of strokes between classes to be similar. Aside from other variables such as student questions, differences in the ability of the students in the different sections of the class, or even the time of day (the later in the day, the more placid the participation) we would expect in the best case scenario to see similar statistics between the same classes. This does seem to be borne out by these graphs, though many more classes would need to be studied to confirm it in a statistically significant way.

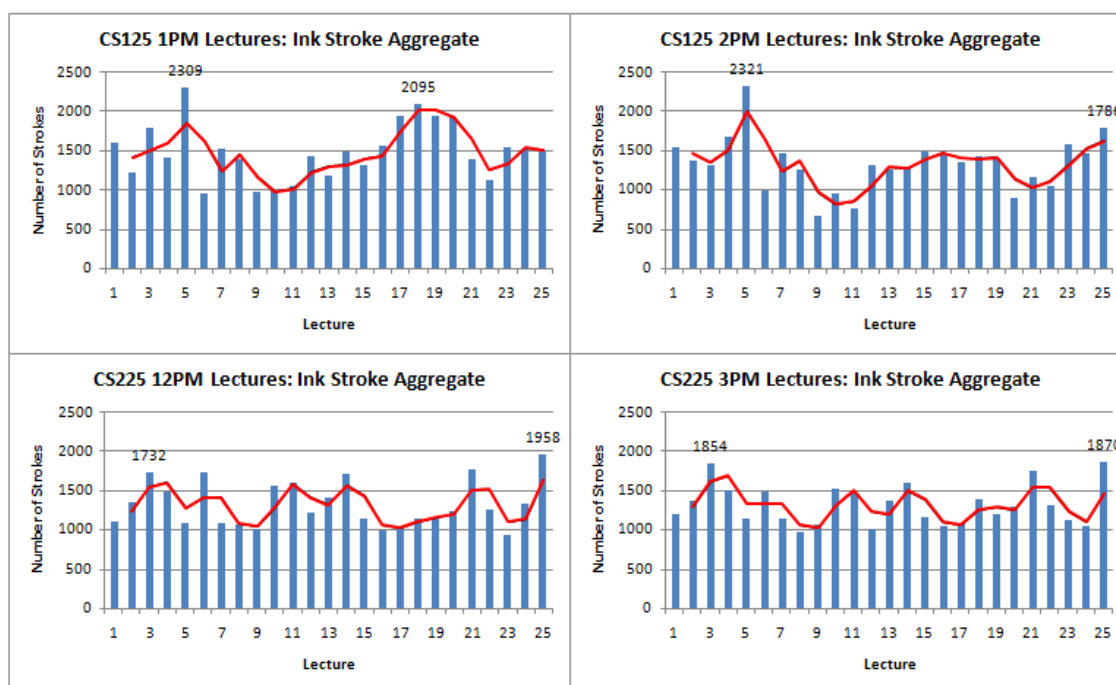


Figure 5.30: A comparison of the average number of ink strokes created per lecture for the spring 2005 CS125 and CS225 lectures.

Total ink strokes deleted

Another example is presenting the number of ink strokes *deleted* within the course of a lecture. Figure 5.31 compares the average number of strokes deleted over the course of each lecture. The earlier section of CS225 has fewer overall deletions per lecture while the opposite is true for the earlier section of CS125. It is difficult to determine why the earlier class of CS225 has fewer deletions unless fatigue became a factor. Although the instructor taught two lectures of another course prior to teaching the 3PM CS225 lecture, the instructor's expertise of the material and experience teaching the course suggests tiredness. Teaching two sections of the same course, one after the other, seems a plausible explanation for the reduction in the number of stroke deletions in the 2PM CS125 class.

Inking time

Inking time refers to the duration of a collection of ink strokes. Recall from figure 5.22 that all ink strokes contain a *total duration*, which is an attribute of Microsoft's ink stroke structure. The total duration is a measurement that reflects how long the digitizer captured the X and Y coordinates when the stylus was detected on the screen⁸. Figure 5.32 shows a comparison of the inking time per lecture for the spring 2005

⁸Although we know that there is a semantic difference between what we call inking time and "writing time" we do not differentiate between the two. In human terms, writing time refers to the duration that a person is putting to ink thoughts or

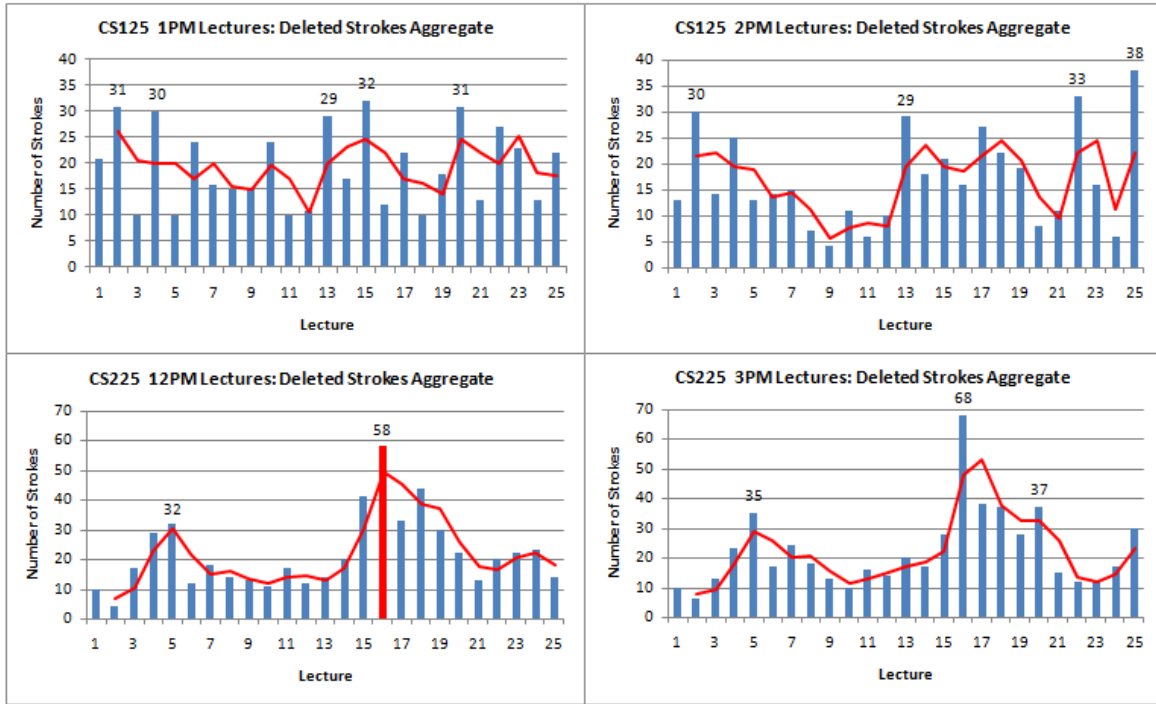


Figure 5.31: A comparison of the average stroke deletion count per lecture for the spring 2005 CS125 and CS225 lectures.

CS225 and CS125 lectures.

Erasure time

The raw number of ink strokes, created and deleted is one way of describing how an instructor interacted with the whiteboard application. Another measurement is the examination of how much time is spent using certain tools. As an example, consider the difference between the *number* of deleted strokes (figure 5.31) and the number of minutes spent *erasing* ink strokes (figure 5.33). A large number of deleted strokes does not necessarily take a greater amount of time. As a case in point, lecture 17 for the 12PM CS225 class shows that the instructor spent nearly one minute erasing ink strokes (the largest duration for the 12PM section of CS225 of the semester). The greatest number of strokes *deleted* in the 12PM CS225 lecture, however, was 58 (see the red bar in figure 5.31) and occurred in lecture 16.

ideas. There are times where we might not be actually writing anything on the page (the pen or pencil might be twirling in one our hands and thus away from the paper as we think) but as long as the writing continues within a certain proximity of time, it would be considered “writing time.” One solution to this semantic incompatibility is to use some constant value representing the maximum amount of time allowed before the “act of writing” is completed.

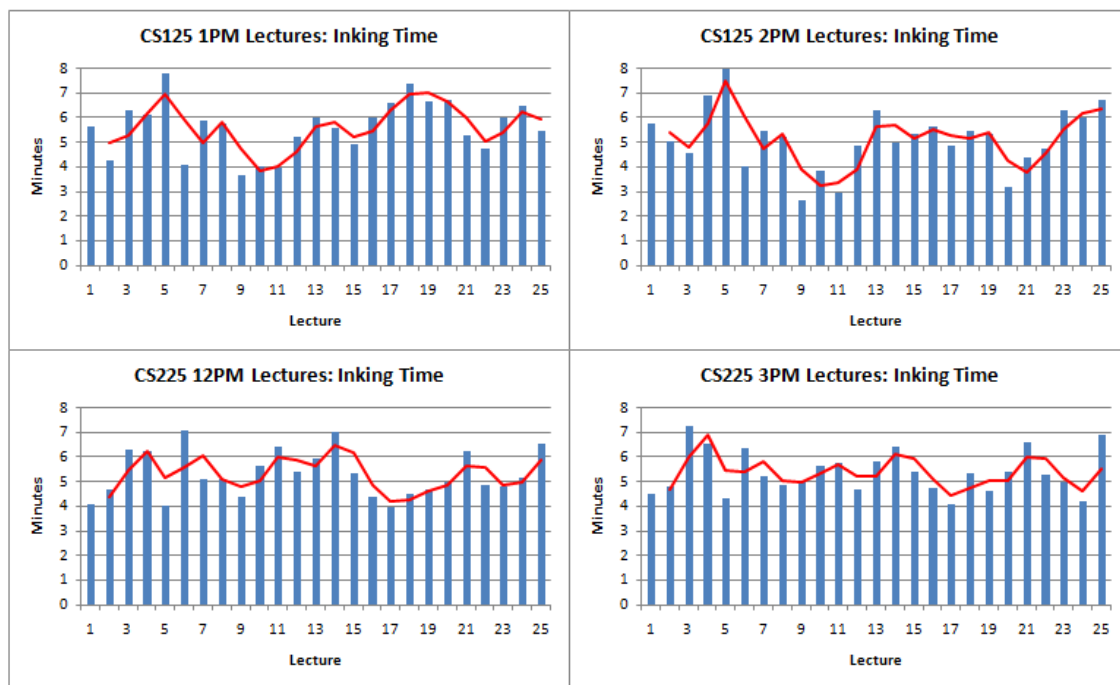


Figure 5.32: A comparison of the inking time per lecture for the spring 2005 CS225 and CS125 lectures.

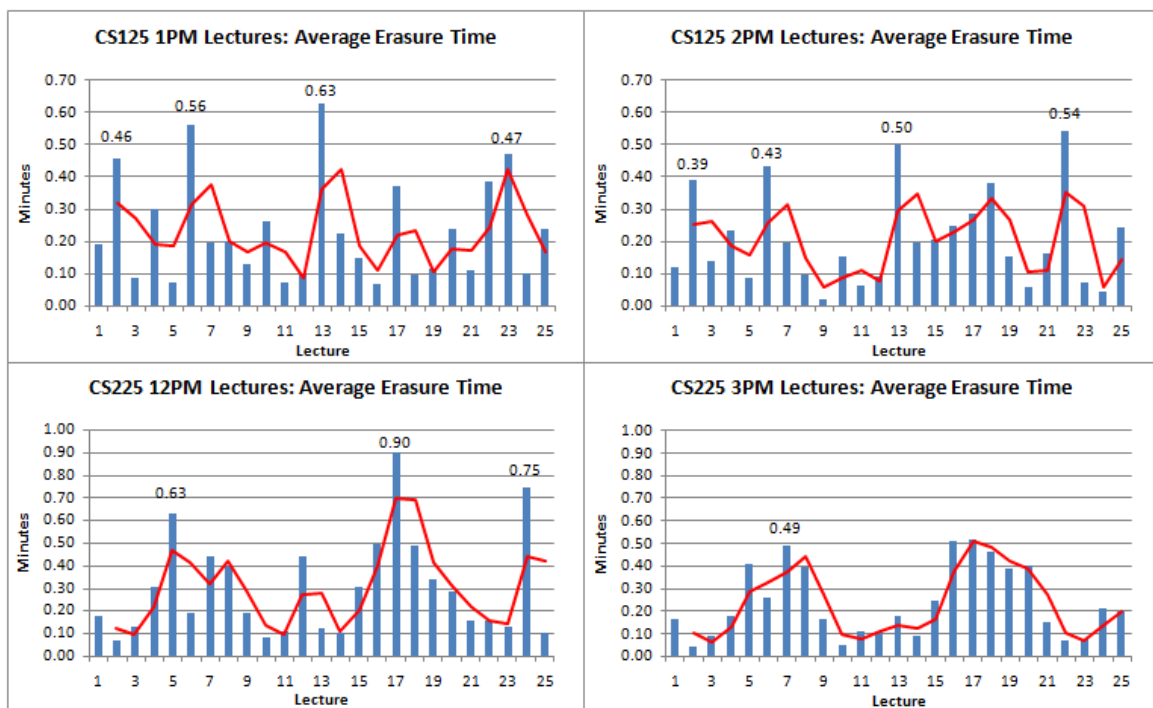


Figure 5.33: A comparison of the average erasing time per lecture for the spring 2005 CS125 and CS225 lectures.

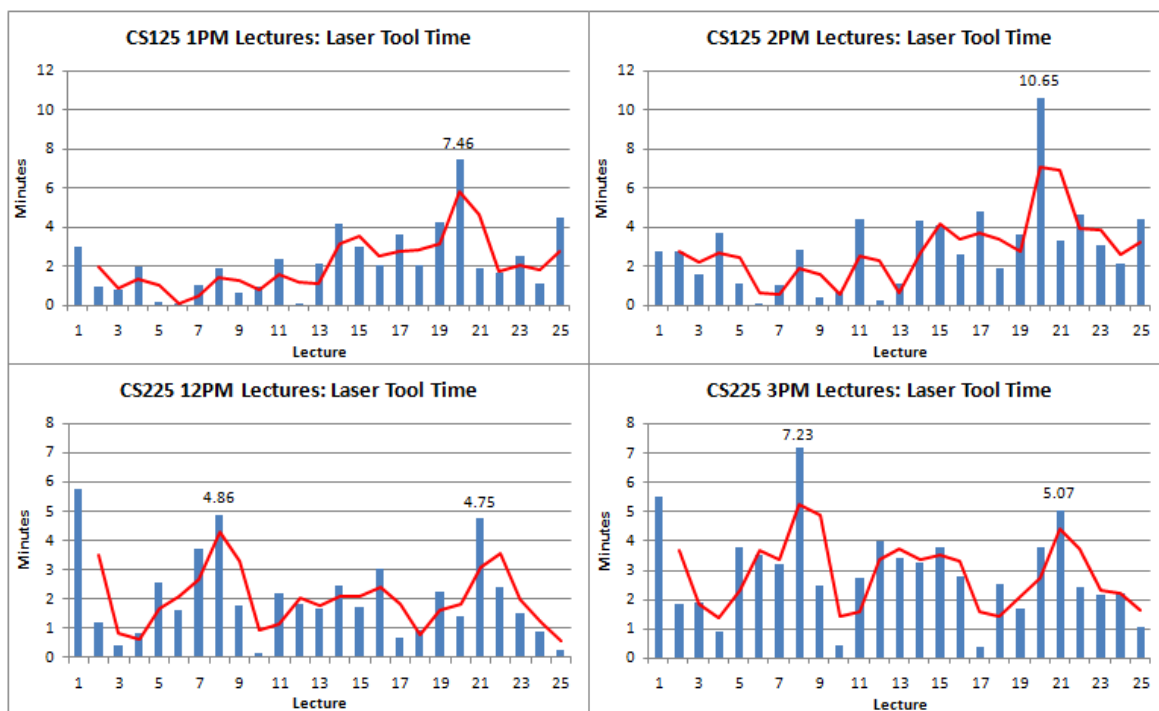


Figure 5.34: A Comparison of the average laser tool time for the spring 2005 CS225 and CS125 lectures.

Laser tool time

The *laser tool* time aggregate (figure 5.34) reports on the number of minutes an instructor spent “using” the laser pointer tool. Having taught both CS125 and CS225, it seems reasonable that the data structures course would have a larger amount of time dedicated to annotation mode, which is what the laser tool provides. When traversing a binary search tree or hash table, the laser pointer is an excellent way to reinforce each step of the algorithm, especially in picture based drawings which dominate in the CS225 course.

Ink stroke speed

We now consider the ink stroke speed (figure 5.35). The average speed is defined by taking the stroke length and dividing it by the timer tick information. The ink stroke length is defined by accumulating the distance between consecutive X and Y points. Recall from figure 5.22 that each ink stroke contains a list of X and Y points, and packet information captured by the digitizer including pressure values and timer tick. The last value in the timer tick list is the total duration in milliseconds of the ink stroke. When we referred to the inking time (see above) that single last timer tick value represents the duration. How fast someone writes could just be a style or if irregular at times, may indicate a “speeding up” affect (like at the tail end of a lecture).

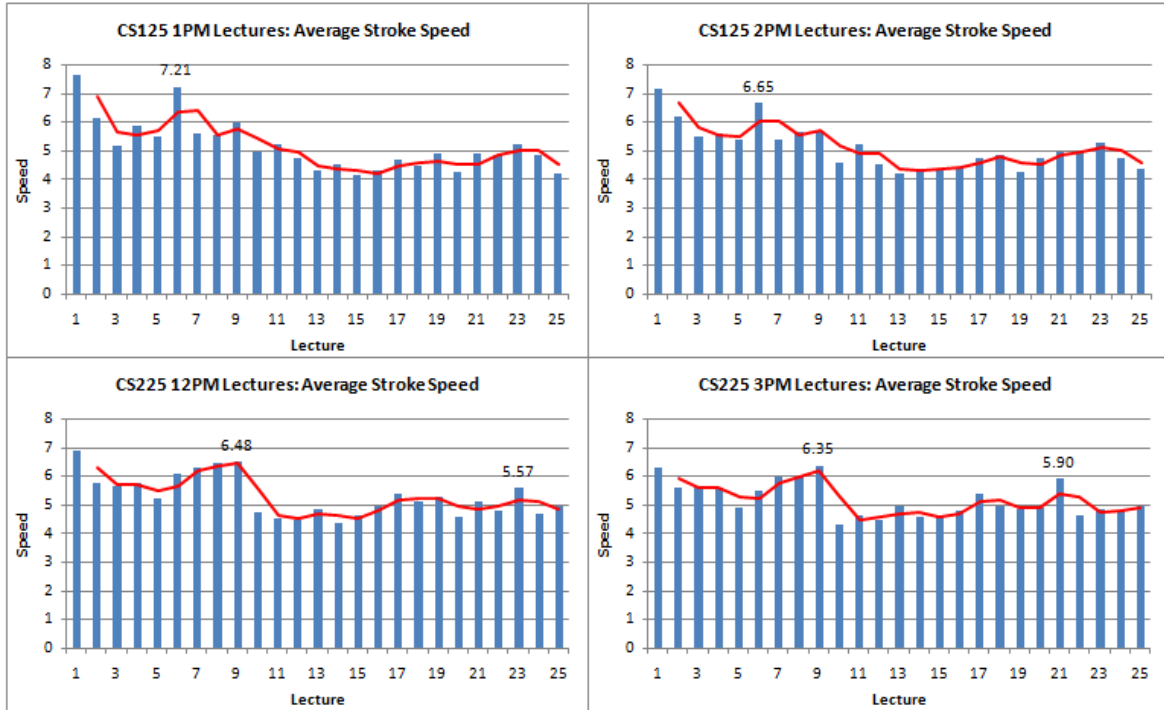


Figure 5.35: A comparison of the average ink stroke speed for the spring 2005 CS125 and CS225 lectures.

Table 5.8: Comparison of spring 2005 and 2006 CS225 lecture attributes

	Total strokes		Total minutes	
	Ink	Laser	Inking	Laser
CS225 Jason Zych				
12PM	48,506	3,678	274.78	75.71
1PM	54,033	5,073	331.47	109.52
CS225 Cinda Hereen				
12PM	41,779	197	205.67	3.78
1PM	44,102	108	218.22	2.64

As a comparison between two instructors (Jason Zych and Cinda Hereen) table 5.8 presents the total number of ink strokes, laser strokes, and time in minutes. The differences in the laser tool usage is striking.

5.5.2 Variations between pages

Ink stroke length

Ink stroke length can help us to determine whether or not the instructor is writing diagrammatically. Shorter length ink strokes are characteristics of code excerpts (i.e. C++ or Java as is taught in CS125 and CS225) while longer ink strokes characterize diagrammatic representations. All things being equal, we would expect similar ink length values for the same classes. Figure 5.36 shows a similar pattern for both of the data structures lectures. As in any teaching, it is possible that the instructor gets “out-of-sync” with one class

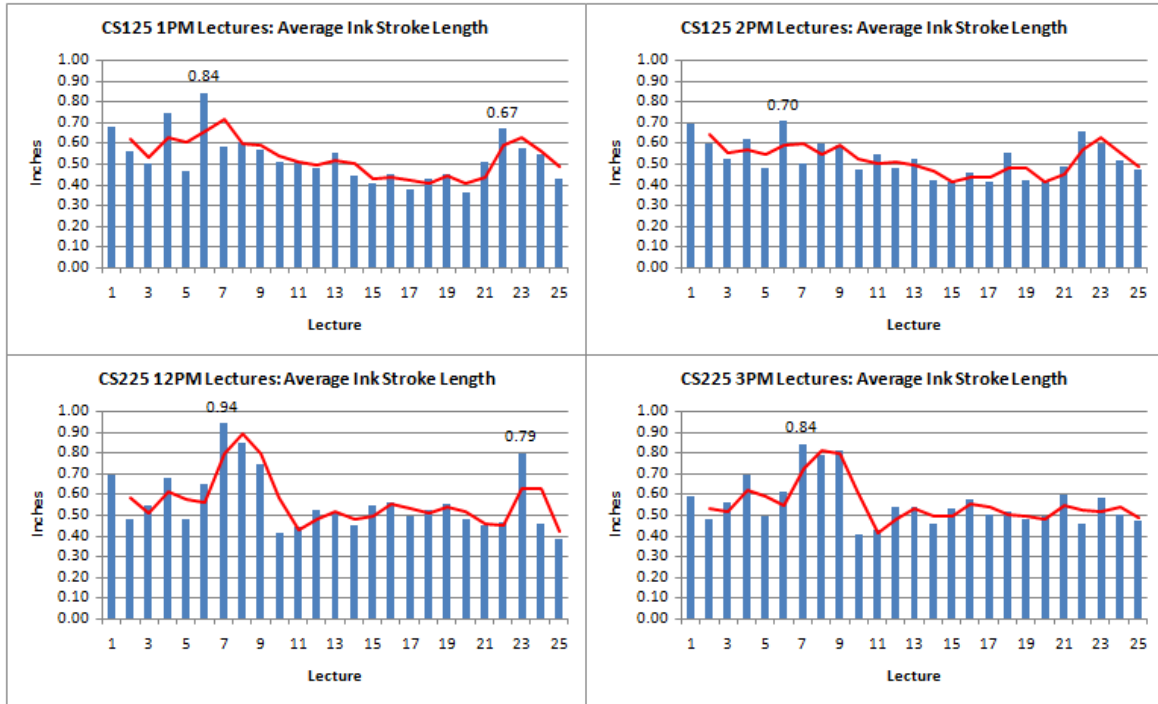


Figure 5.36: A comparison of the average ink stroke length for spring 2005 CS125 and CS225 lectures.

by falling behind in another. There is a lot of material taught in both of the core courses taught by Jason, and it would not be surprising if students asked more questions in one class then another.

There were times where the exported PDF file did not coincide with the final state of the log files. For example figure 5.37 illustrates a graph of ink density (defined as the total amount of ink quantity in inches). Note the hollow bar graphs for pages 0 to 19 inclusive. At some point after the lecture “ended,” the instructor deleted all but the last 3 pages of the document. In our log file analysis, we wanted to be consistent amongst lectures to limit the analysis of application interaction that occurred only during the 50 minute teaching period for each lecture. Jason was meticulous about finishing and beginning lectures on time. Although he continued to address student questions during the 10 minute period before the next lecture was to begin, we did not want to “count” his interaction with the whiteboard application. Jason did two things with a high level of consistency during his 10 minute breaks: deleted pages in the process of editing the lecture document to be posted on the course website, and used the laser pointer to answer student questions.

Classification of pages in CS125 1PM lectures

In an attempt to characterize pages as being either mostly text-intensive or picture-intensive, we classified 230 pages from the CS125 1PM class. Each page was assigned an integer value in the range of one to

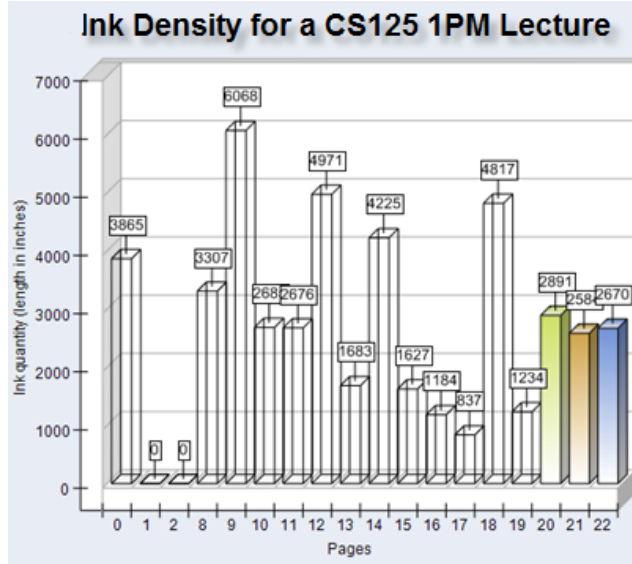


Figure 5.37: Measurement of ink density

Table 5.9: Ink stroke *length*, *pressure*, and *speed* values for pages characterized as diagrammatic

Rank	Length		Pressure		Speed	
<i>value</i>	<i>avg</i>	<i>stdv</i>	<i>avg</i>	<i>stdv</i>	<i>avg</i>	<i>stdv</i>
1	0.33	0.12	138.34	42.14	4.15	1.56
2	0.47	0.17	154.93	13.21	5.10	1.15
3	0.52	0.21	153.83	12.68	4.94	0.90
4	0.63	0.22	148.74	12.24	5.37	1.24
5	1.00	0.56	145.37	14.49	6.27	2.02

five where one contained virtually no diagrammatic⁹ attributes and five contained almost all diagrammatic attributes. Figures 5.38, 5.39, 5.40, 5.41, and 5.42 illustrate example pages assigned to the five groups respectively.

The results, presented in table 5.9, show a strong correlation between the classification and average stroke length. Pages classified as highly diagrammatic have a larger average length.

Annotation versus presentation

Classroom teaching generally consists of alternating periods of pure exposition and elaboration (explanation, question-answering). When using prepared slides, the instructor will first go over the slide and then explain it further, possibly with ink annotations, use of the laser pointer, adding new pages for further examples, and so on. When not using prepared slides, one will normally see, in post hoc analysis, a line of thought organized in a more or less consecutive stream, and then annotations clearly added afterward. Can we determine how

⁹Diagrammatic is defined as "...a plan, sketch, drawing, or outline designed to demonstrate or explain how something works or to clarify the relationship between the parts of a whole." <http://www.thefreedictionary.com/diagrammatic>

<p>for flow of control all you need is.</p> <ol style="list-style-type: none"> 1) sequential execution 2) conditional 3) loop 	<p>Infinite Loop</p> <pre>while(true) { ; }</pre> <p>CTPL-C</p> <pre>int i=1; int c=7; while(i<=10) { c--; // i++; c++; }</pre>
<pre>void KnockTask(int n)/n>=0 { if (n==0) base case Knock(); else recursive case { TakeStep(); KnockTask(n-1); } }</pre>	<p>Recursion</p> <p>language of algorithm design</p> <p>sometimes loop is implementation technique</p> <p>easier to understand hard algorithms when explained recursively easier to prove correct</p> <p>defining a problem's solution in terms of a solution to similar- but-smaller problems</p>

Figure 5.38: Pages assigned a diagrammatic ranking of one are almost all textual

<pre>int Search(int[] arr, int key)</pre> <p>search array pointed to by arr for key; if there, return index where it is located; else return -1</p>	<p>Encapsulation</p> <p>↳ hiding a class's implementation behind an interface</p> <p>client of class ↳ forced to use interface</p>
<p><u>Arrays</u></p> <pre>int total=0; int i; for (i=1; i<=10; i++) { Supn ("Enter value" + i + ": "); int value = Keyboard.readInt(); total = total + value; } Supn (total / 10.0); total = total + <u>Keyboard.readInt();</u></pre> <p>120 84170 89771</p> <p>Enter value 1: Enter value 2: Enter value 3:</p>	<p>Why???</p> <p>objects <u>not</u> bound by scope</p> <p>↳ objects reclaimed by system once your program has <u>no</u> reference variables pointing to that object</p>

Figure 5.39: Pages assigned a diagrammatic ranking of two are mostly textual with a few annotations such as underlines and question marks

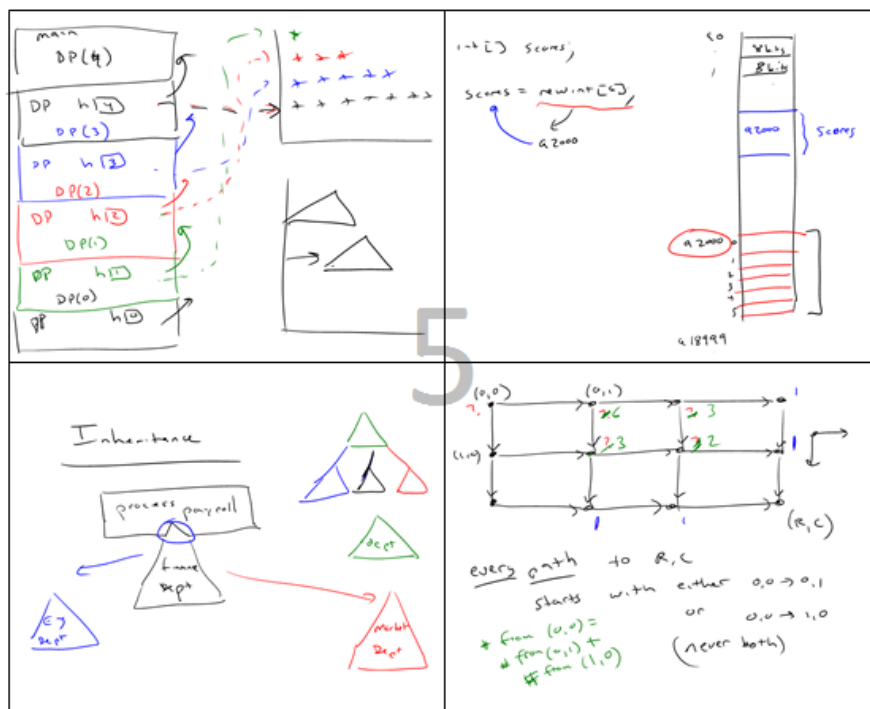


Figure 5.42: Pages assigned a diagrammatic ranking of five contain almost all diagrammatic attributes

much time was spent in basic exposition (creation mode) and how much in explanation (annotate mode)? Is the ratio consistent across classes for a single instructor? Does it change during the day or through the semester? And can we compare the amounts of time in each activity between an instructor using prepared slides and an instructor writing from scratch? We need to give more precise definitions to these terms, and produce an algorithm (which will necessarily be approximate), but we believe this data is interesting as an indication of the basic pedagogy employed by each teacher.

There are a number of different ways we might distinguish annotation from creation of pages. For example, whenever an insertion of a blank page occurs, while the instructor is writing on that page in a “top-down” fashion, we can assume that the instructor is in writing mode. Another heuristic for determining whether an instructor’s ink strokes at any point in time are annotation or writing is if the page is a re-visitation. Regardless of the reason an instructor traverses to an already-created or already-viewed page, emphasis is placed on that page, and if the instructor writes on the page (including laser tool strokes), it is considered annotation. If the instructor does not write (annotate) on the page but a visitation is recorded for the page, he or she may have articulated a point without interacting with the whiteboard application. Unfortunately, without recording an audio feed, we would not be able to be exactly sure that the visitation was an annotation even though emphasis may have been placed on content by the instructor’s voice.

Histogram of Stroke Speed for CS225 Spring 2005

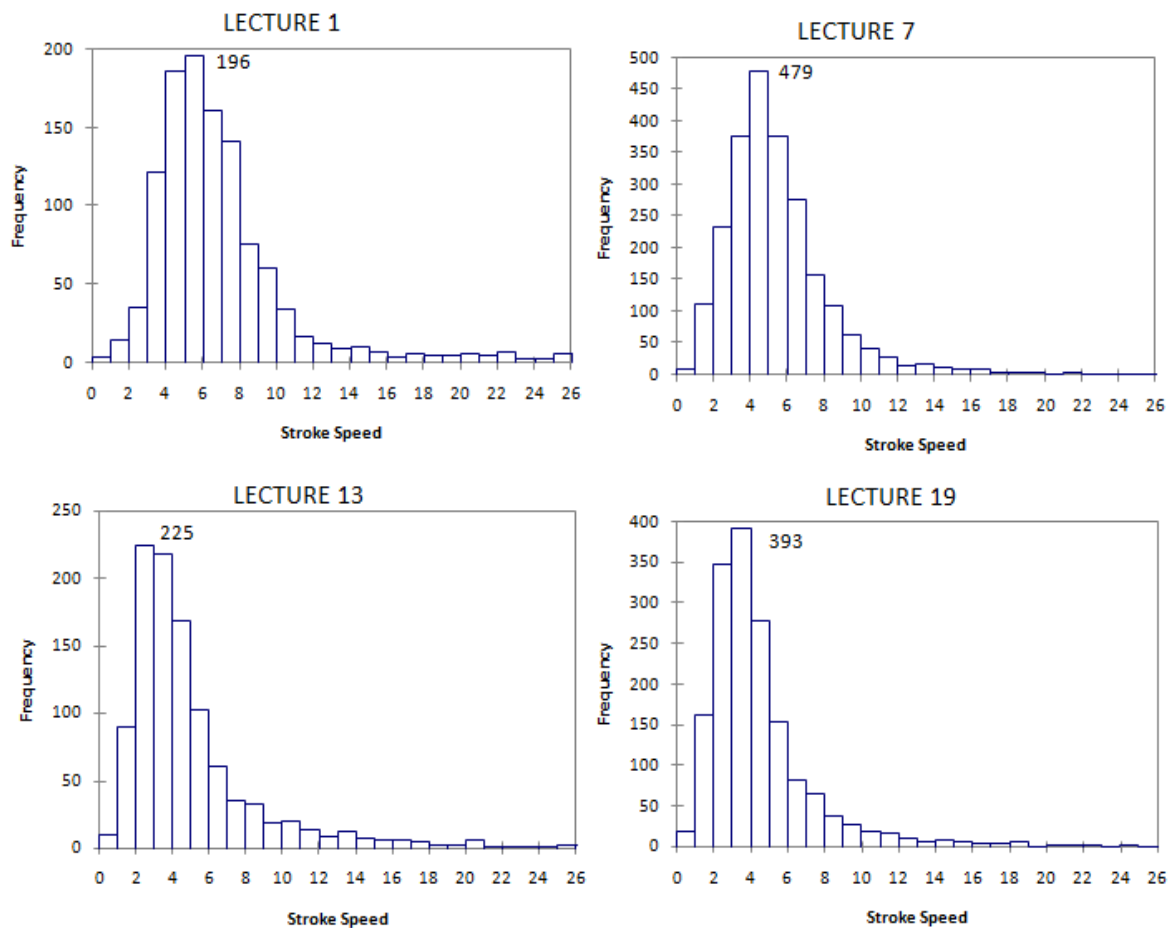


Figure 5.43: A comparison of ink stroke speed for four lectures of CS125 in the spring 2005 semester.

5.5.3 Comparison between classes

Do the measures taken above change in some consistent way during the course of a lecture or subsequent classes? We might take this as a measure of fatigue (something one might be interested in being able to determine in real time). When processing the log files, not only did we have enough information to replicate exactly what happened in the application, we also had the ability to easily accumulate features that might help us to answer interesting questions. Since Jason taught the same course twice per day, it was reasonable for us to assume that there might be similarities between lectures. All things being equal, one would expect the instructor to repeat much of what was presented in the previous lecture. Although this hypothesis seems obvious, we wondered if the data could in some way illustrate such similarities. Figure 5.43 gives a comparison of ink stroke speed for lectures 1, 7, 13, and 19 of CS125. While it appears as if the instructor wrote at varying speeds, the later lectures (13 and 19) seem to suggest a “slowing down” of ink stroke speed.

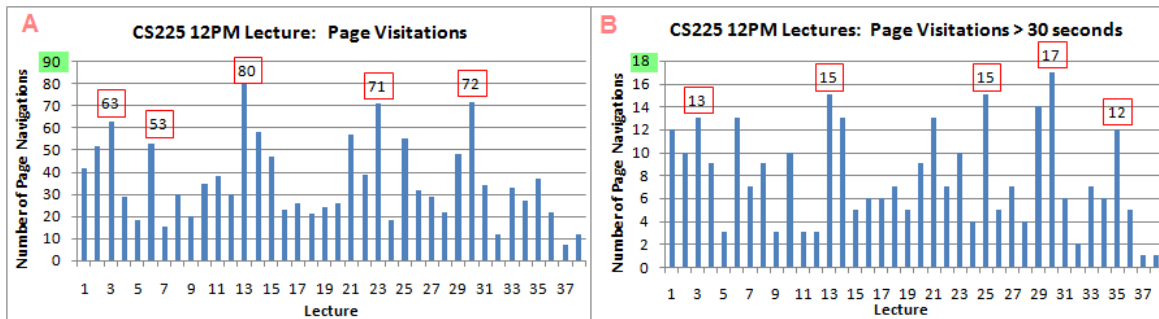


Figure 5.44: Page navigations in the spring 2005 CS225 12PM lectures

Page visitation

Whenever the instructor traversed to a previously visited page we recorded the time and duration of the visit. Page visitation refers to the number of times an instructor navigates away from a page and at some later point in the lecture, returns to the page. Since the whiteboard application provides a linear means of navigating the lecture document, we needed to set a minimum time value required to stay on a page so that we did not confuse the navigation through a page with the arrival at the destination page. For example, if an instructor was on page 5 of their document and they navigated using the “previous page button” four times (page 5, page 4, page 3, page 2, page 1), we would not want to count the pages en route to the destination page, page 1. Thus, we required that an instructor remained on the destination page for at least 30 seconds. Part A of figure 5.44 shows the total number of page navigations, up to 80 in one lecture. When we added the constraint that a page visitation was meaningful if and only if it lasted at least 30 seconds, a more reasonable number of page visitations resulted (part B).

Aside from knowing which slides were emphasized by the instructor using the laser pointer tool, we might also want to know which pages (if any) the instructor revisited. A revisit could indicate either a question from a student on a previous slide (which would be beneficial for the instructor to know for future class sessions) or that an instructor felt the need to reinforce content from a previous slide. Reinforcing content (even if it is a correction) constitutes annotation. So page navigation can help with the deciphering of what ink strokes are annotations and which are writing (creation). Figure 5.45 illustrates two different lectures where the instructor revisited a number of pages. The duration (in minutes) of each visit is indicated by the number labels on the stacked bars; the higher the bar (in terms of how the bars are stacked), the later the visit in the lecture.

If an instructor was teaching a lecture and as a result of student questions throughout the course of the class session needed to resort to a previous page to explain an objective in the lesson, we would expect to see additional page visitations to that source page. An instructor might adapt their teaching to spend more time

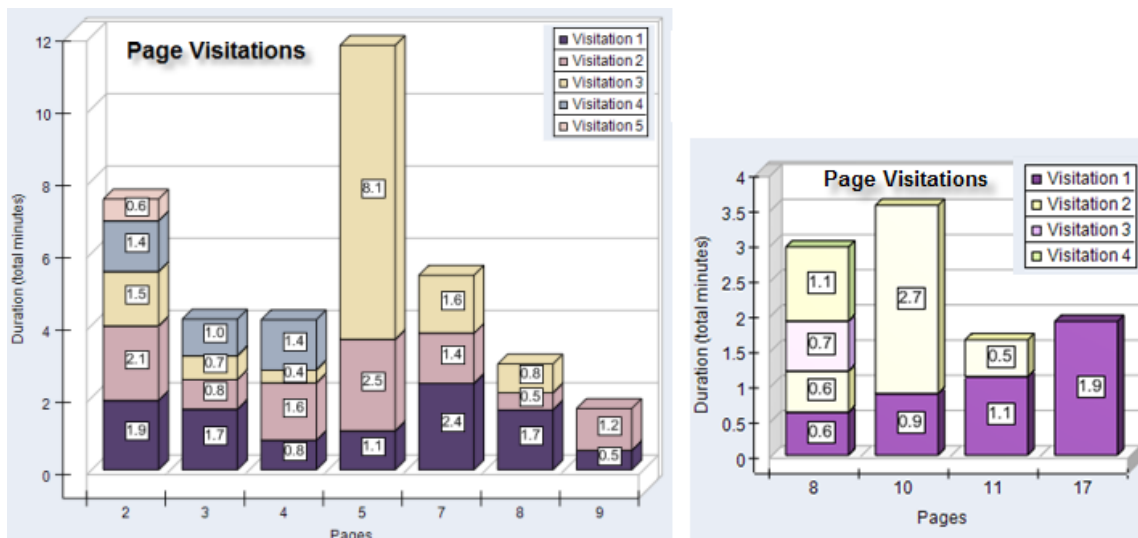


Figure 5.45: Page visitations for two lectures

or integrate additional explanations while on that “problematic slide” for future class sessions. As a result, we would expect to see fewer page visitations in the subsequent lecture. Consider as an example the graphs in figure 5.46, which show page visitations in two lectures on the same day (January 31, 2005). On the whole, the earlier section of CS225 (12PM) contains more page visitations—six pages were revisited—than the 2PM lecture, which only has two pages that were revisited. Page three of the 12PM lecture had five total visitations (four being “revisits”) lasting 3.1, 0.7, 1.0, 0.6, and 1.6 minutes respectively. The third page of the 2PM lecture had one less revisit, but the total time spent on the page is also less than the earlier lecture. If we assume that the lecture documents contained the same content, it would be interesting to investigate why there were more visitations in one than in another.

Note that the values in figure 5.46 do not differentiate between *inking time* and *spectator time* (time page was displayed but no inking occurred). The graph gives the *total* time spent on the page for each visit. An instructor might spend significantly more or less time on certain pages for one class period than another. While rationales for such differences can be many, the ability for us to easily capture and query this information can provide insight into an instructor’s time management skills, the different levels of the students in different sections (or lectures) of the same class, or at the very least, an indication of which pages of a lecture document (that will be posted for students to download after the lecture) an instructor spent the most amount of time on. A categorization of total time in terms of “inking time” and “spectator time” in minutes per page is presented below.

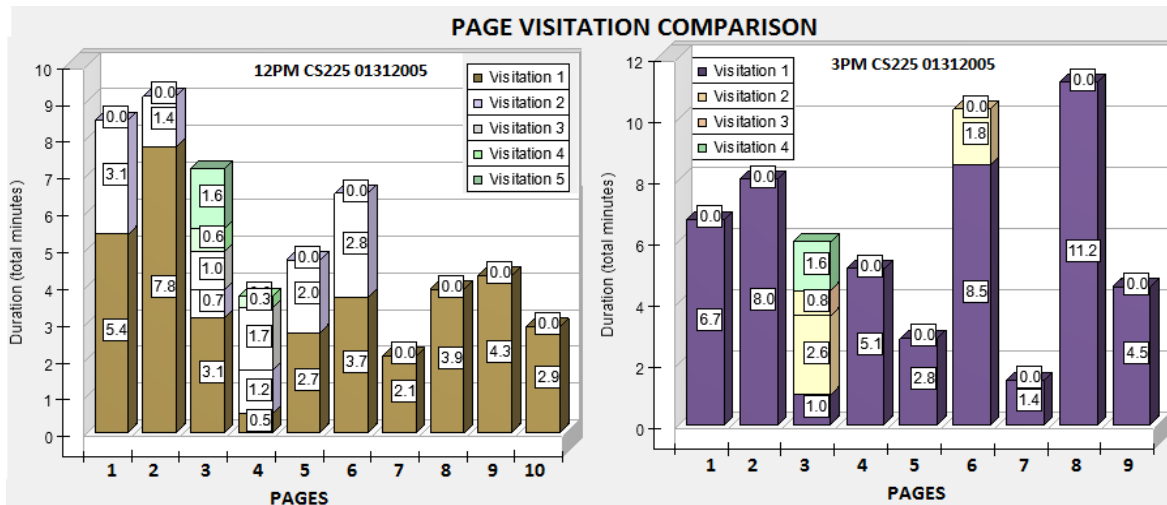
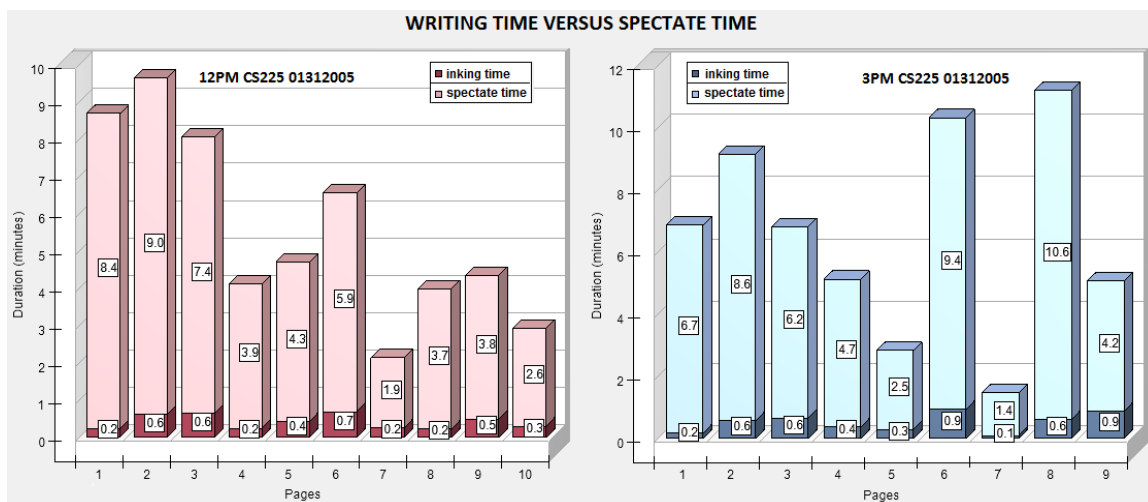


Figure 5.46: A Comparison of page visitations in two CS225 lectures







In the next two sections we present summary information for two lectures: CS125 lecture 11 and lecture 23 for both CS125 and CS225.

CS125 Lecture 11: February 11, 2005

The summary information for both of the CS125 lectures is presented in figure 5.47 and the corresponding PDF document is displayed in figures 5.48(pages one to four) and 5.49 (pages five to nine). Once again, the pages have been juxtaposed to illustrate pages containing the same lecture material.

You might note that the table summary for the 1PM lecture contains an extra page in the lecture document; the table shows that the 1PM lecture has seven pages and the 2PM lecture, nine pages. This is correct. The summary table does not reflect the fact that an instructor may have deleted a page during or after the lecture. The statistics collected and presented in this chapter are all reporting on what happened

LECTURE	NUMBER STROKES	INK LENGTH	AVG. PRESSURE	AVG. SPEED
2/11/2005 1PM				
1	154	3664.67	173.97	7.96
2	80	1813.47	164.10	9.12
3	50	1083.59	169.37	7.93
4	219	10413.31	171.19	7.83
5	154	2657.09	171.48	6.20
6	187	2559.38	158.73	5.60
7	135	5238.34	157.13	8.59





LECTURE	NUMBER STROKES	INK LENGTH	AVG. PRESSURE	AVG. SPEED
2/11/2005 2PM				
1	102	2587.68	173.12	8.03
2	54	893.35	161.33	6.79
3	56	1281.16	170.48	8.82
4	68	992.03	164.00	7.24
5	60	1113.42	159.64	7.71
6	15	369.99	170.38	8.06
7	194	9103.08	166.60	7.29
8	135	2321.30	174.35	6.09
9	215	2384.73	160.71	4.99

Figure 5.47: Summary information for CS125 lectures for February 11, 2005

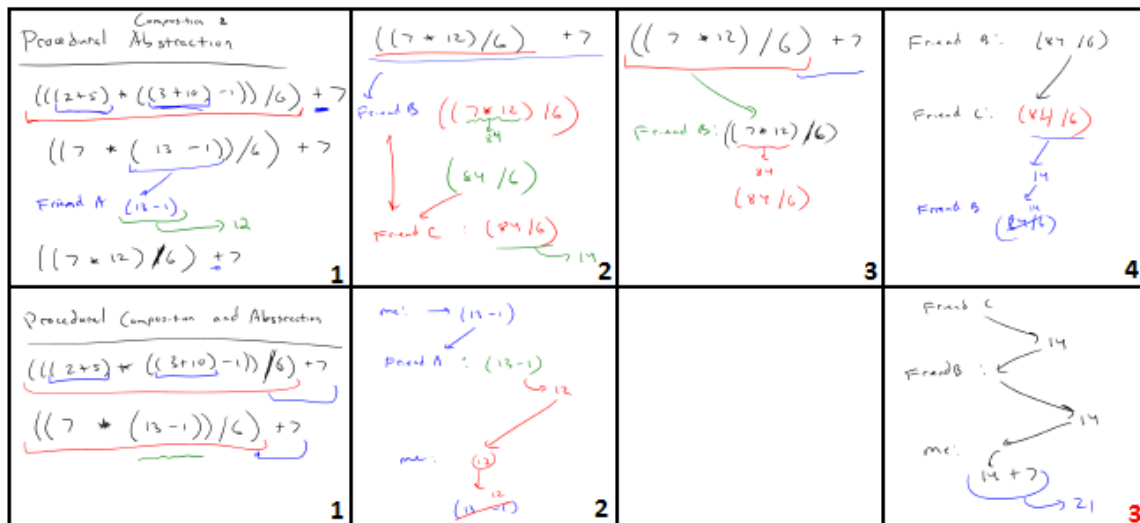


Figure 5.48: Pages one to four of the CS125 whiteboard documents for February 11, 2005

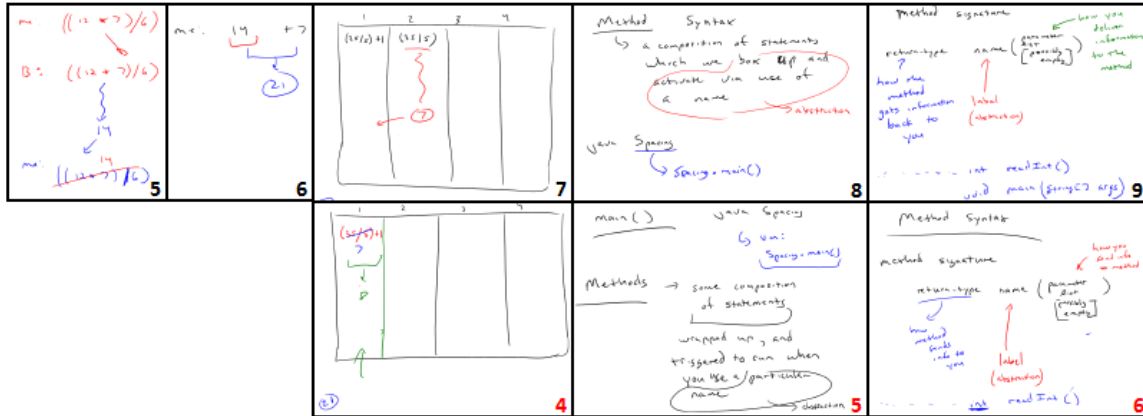


Figure 5.49: Pages five to nine of the CS125 whiteboard documents for February 11, 2005

during the 50 minute lecture, not before, and not after. The summary information provides a rough overview of what transpired in the whiteboard application for the duration of the lecture. In the case of the 1PM lecture, the log files show that the instructor *deleted* a page of the document during the lecture. From our analysis of the 2005 spring semester, the deletion and editing of pages almost always occurred after the lecture had already ended. The rationale was to edit the PDF document for posting to the course website. To further illustrate we present the total time allocation per page graph for the 2PM CS125 lecture (figure 5.50).

CS125 Lecture 23: March 14, 2005

Let us consider a comparison of one lecture date to see if similarities exist between the same courses. CS225 was taught at 12PM and 3PM and CS125 was taught at 1PM and 2PM. We will look at the CS125 and CS225 lectures for March 14, 2005.

The two lectures of CS125 contain the same number of pages (figure 5.51). We have added a column for reporting the number of laser pointer strokes per page (the last column). Note the difference between the *number of strokes* and the average *ink length* attributes. Larger values in the ink length column are indicative of the creation of longer strokes. The page containing the second largest average ink stroke length (and containing more than 100 ink strokes) is page eight in both of the documents. The page three ink length for the 1PM lecture (3053.85) and 2PM lecture (1969.00) are larger than in page eight, however, the number of strokes is *much* smaller (25 each) and characterizes a smaller sample size. One other curious value is the number of strokes for page one in the 1PM lecture. With only one stroke, and seventeen laser pointer strokes, it is clear that this page of the presentation already contained content (from either a previous lecture, or prepared just before the lecture began). The log file indicates that the instructor copied

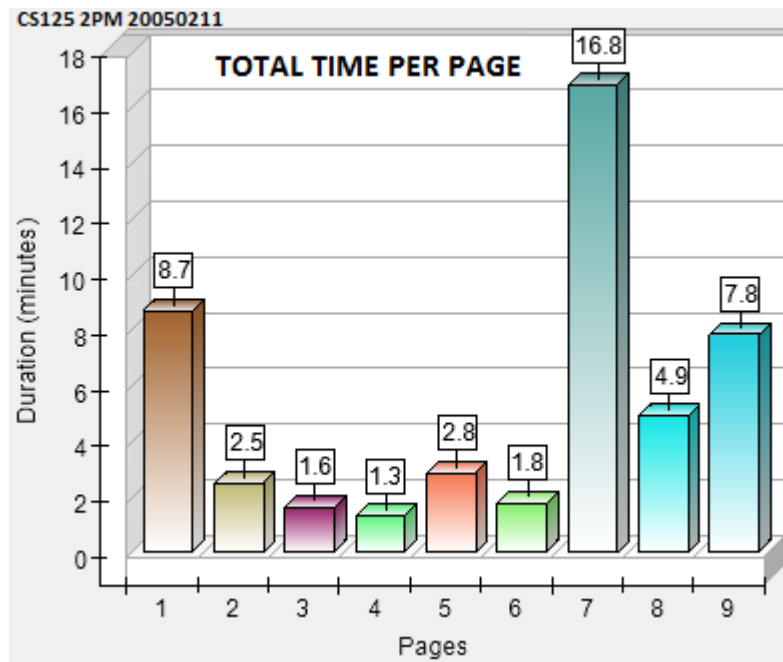






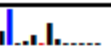
Figure 5.50: Total time allocation for the 2PM CS125 lectures on February 11, 2005

approximately 84 ink strokes from a previously created document (lecture 22) and pasted the contents into the first page of the newly created document (lecture 23). The corresponding PDF documents are illustrated in figures 5.52 and 5.53. The documents are nearly identical with small differences in ink stroke color and additional diagrammatic attributes on pages one, five, six, and eleven.

Figures 5.54 and 5.55 show the page visitation and total time spent on each page for the two lectures. Note that the second page in each document contained three visitations. The first two pages of each document is where the instructor spent the most time. Perhaps the increase in time spent on the first page of the 2PM lecture (as compared to the first page of the 1PM lecture) is a result of the instructor drawing from scratch rather than referring to already-created whiteboard content (which according to the logs for this particular lecture, was most certainly the case).

CS225 Lecture 23: March 14, 2005

Let's now look at a comparison of the two CS225 lectures for the same date summarized in figure 5.56. What is immediately curious is the creation of 60 laser pointer strokes on page one in the 3PM lecture versus 22 in the earlier, 12PM lecture. A close look at the documents created during the CS225 lectures (figure 5.57) shows the addition of page five in the 12PM lecture (which according to the summary table contained one extra page). The juxtaposition and numbers assigned to the pages accentuate the addition of a single page in the 12PM lecture (page five). The rest of the pages are paired as visual complements between the two

LECTURE	NUMBER STROKES	INK LENGTH	AVG. PRESSURE	AVG. SPEED	LASER STROKES
3/14/2008 1PM					
1 Lecture 23a	1	1130.78	185.09	1.91	17
2	176	1566.79	149.76	3.74	47
3	25	3053.85	172.73	6.36	1
4	108	960.97	150.33	4.11	4
5	84	1561.12	146.94	5.56	13
6	85	1463.54	154.80	4.86	0
7	170	893.79	159.07	3.64	28
8	137	1824.69	151.25	4.77	6
9	73	1151.65	143.43	4.90	0
10	85	948.91	150.89	3.81	0
11	115	1823.15	150.03	3.67	0
12	66	1288.29	156.12	3.65	0
13	50	1491.06	149.85	5.25	0





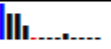
LECTURE	NUMBER STROKES	INK LENGTH	AVG. PRESSURE	AVG. SPEED	LASER STROKES
3/14/2008 2PM					
1 Lecture 23b	156	1127.00	132.00	4.00	17
2	166	1721.00	148.00	3.90	12
3	25	1969.00	165.00	3.80	12
4	119	926.00	149.00	3.80	6
5	73	894.00	145.00	4.80	0
6	67	928.00	145.00	4.10	0
7	179	1243.00	150.00	3.80	0
8	139	1838.00	145.00	5.10	0
9	83	1093.00	141.00	4.40	2
10	78	973.00	153.00	3.50	0
11	94	1648.00	159.00	4.40	0
12	64	1476.00	176.00	4.40	0
13	22	1319.00	151.00	6.20	0

Figure 5.51: Summary information for CS125 lectures for March 14, 2005

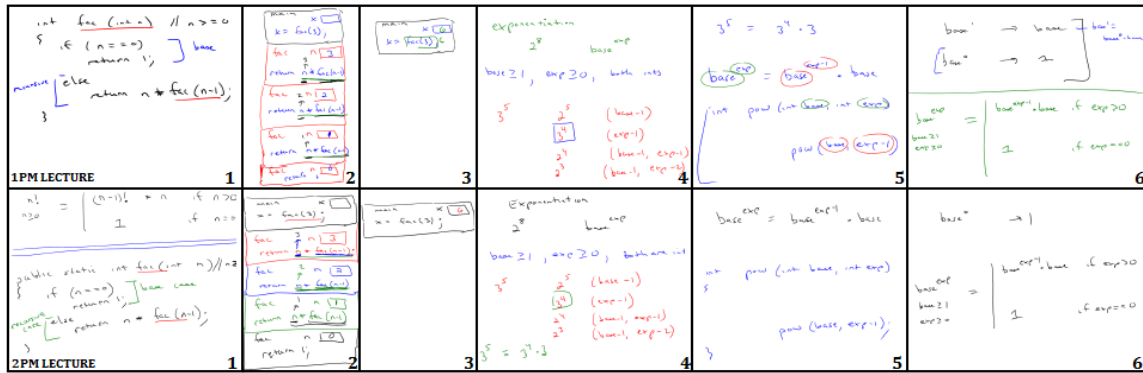


Figure 5.52: Pages one to six of the CS125 lecture whiteboard documents for March 14, 2005

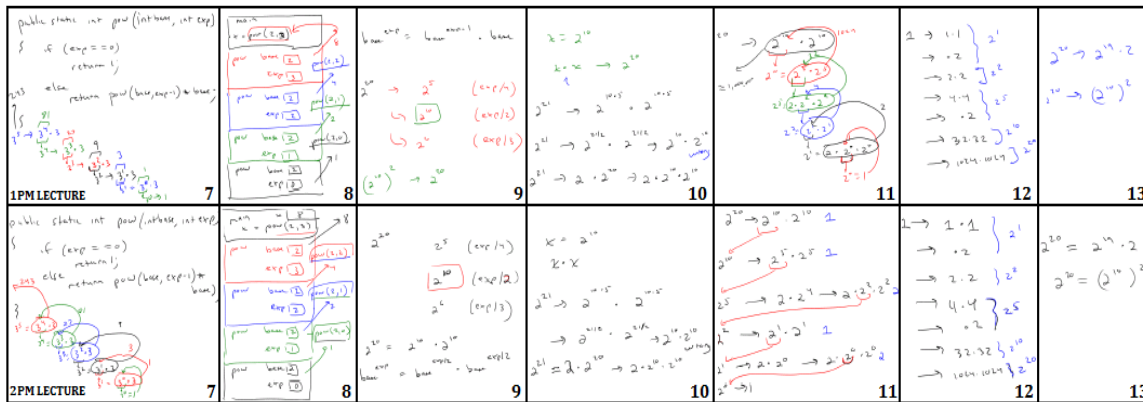


Figure 5.53: Pages seven to thirteen of the CS125 lecture whiteboard documents for March 14, 2005

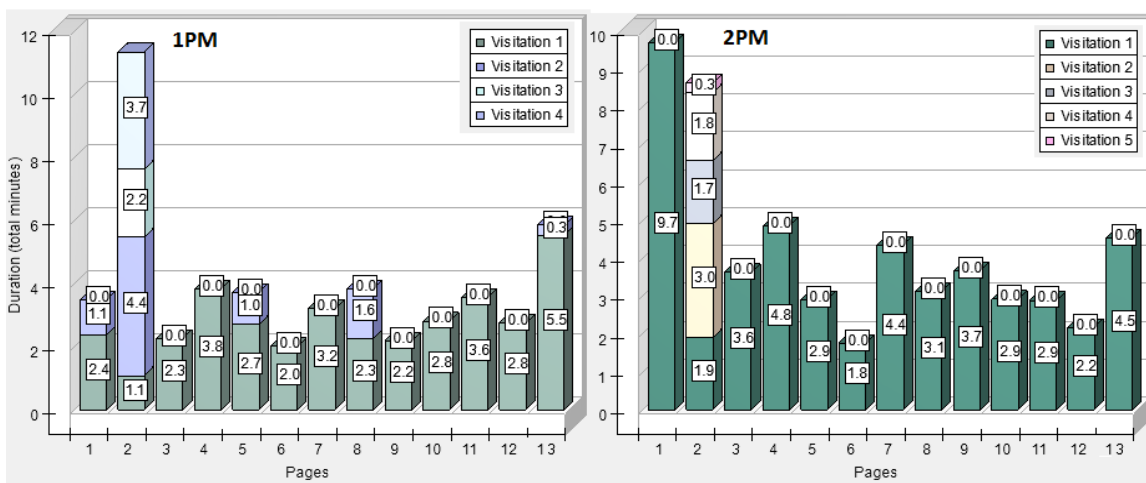


Figure 5.54: Page visitation for CS125 lectures on March 14, 2005

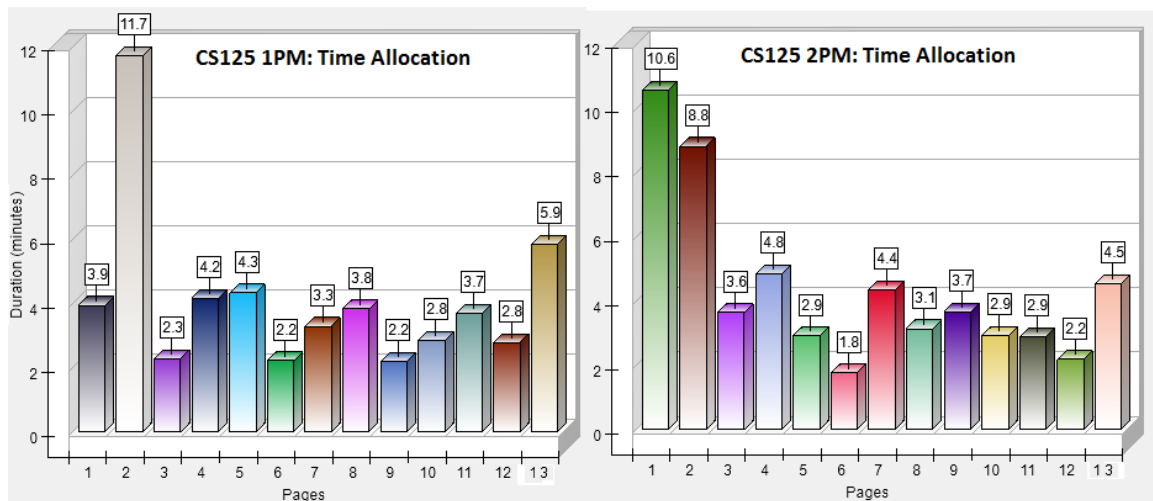


Figure 5.55: Total time allocation for CS125 lectures on March 14, 2005

LECTURE	NUMBER STROKES	INK LENGTH	AVG. PRESSURE	AVG. SPEED	LASER STROKES
3/14/2008 12PM					
1 PAGE	360	1084.43	155.58	3.93	22
2	129	1409.23	146.13	4.51	23
3	58	1648.46	149.31	5.69	0
4	27	2750.29	141.48	8.17	0
5	102	1267.93	139.57	5.60	13
6	129	964.08	152.36	4.58	0
7	135	1072.52	150.24	4.87	13
8	51	2120.65	143.84	7.58	0
9	273	1038.61	152.22	4.18	0
10	51	2413.15	140.33	5.87	0
11	98	2043.40	145.17	6.26	45

LECTURE	NUMBER STROKES	INK LENGTH	AVG. PRESSURE	AVG. SPEED	LASER STROKES
3/14/2008 3PM					
1 PAGE	394	1141.15	152.20	3.94	60
2	145	1324.77	142.13	4.86	48
3	62	1740.35	142.02	5.92	8
4	133	1192.50	136.86	4.89	0
5	67	1098.88	157.40	4.92	0
6	137	1292.46	145.16	5.18	7
7	59	1956.93	151.06	7.17	7
8	244	1209.73	156.78	4.57	0
9	48	2500.23	150.15	6.72	10
10	82	2087.11	156.28	6.89	0

Figure 5.56: Summary information for CS225 lectures for March 14, 2005

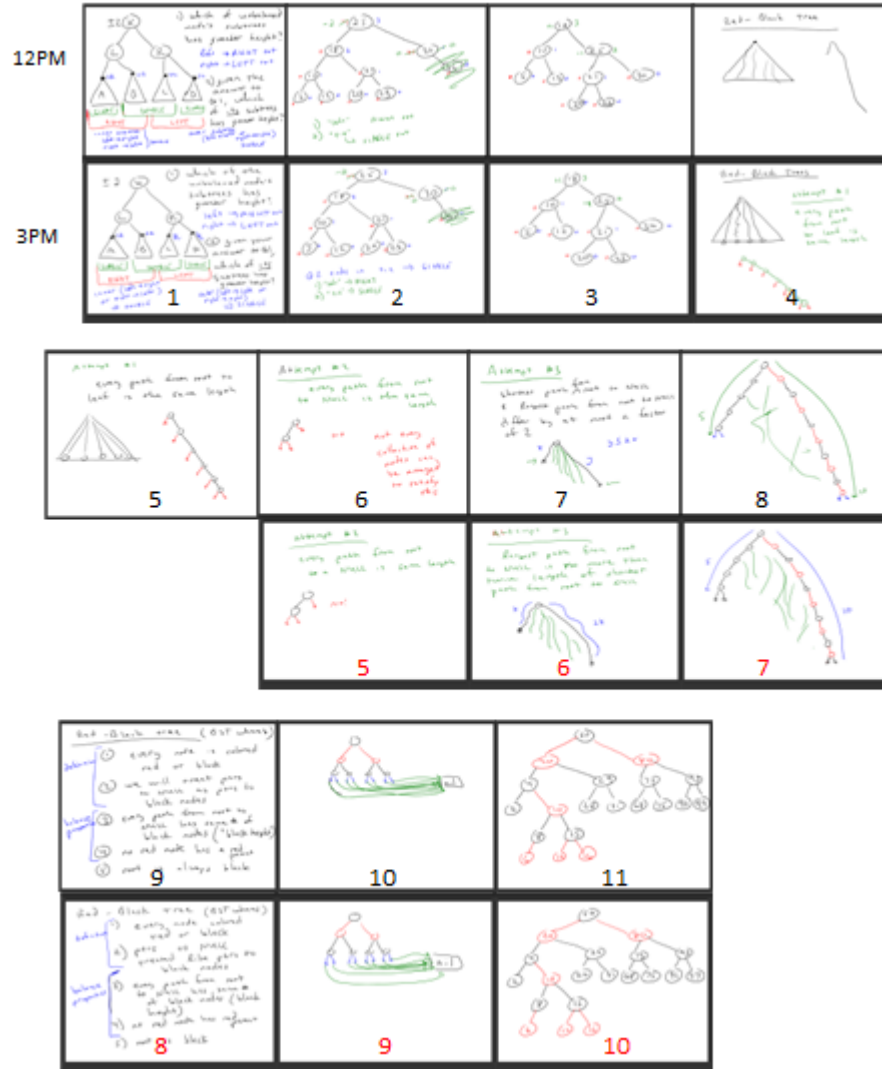


Figure 5.57: CS225 whiteboard documents for March 14, 2005

lectures. The writing speeds for the two documents is similar, with the exception of page four (figure 5.58).

5.5.4 Changes over time for the semester

Did the instructor change how he or she wrote, in a consistent way, during the semester? This would be interesting because we would have no reason to think that such changes were specific to the Tablet PC. Did his or her use of the whiteboard application change over the course of the semester (figure 5.59)? Such changes could indicate that the instructor was learning to use the tool better, and could provide feedback on what features were most useful.

There are other features that could also be used to shed some light on an instructor's style of teaching. For example, the measurement of time lapses between the insertion of a new page, the time that elapses

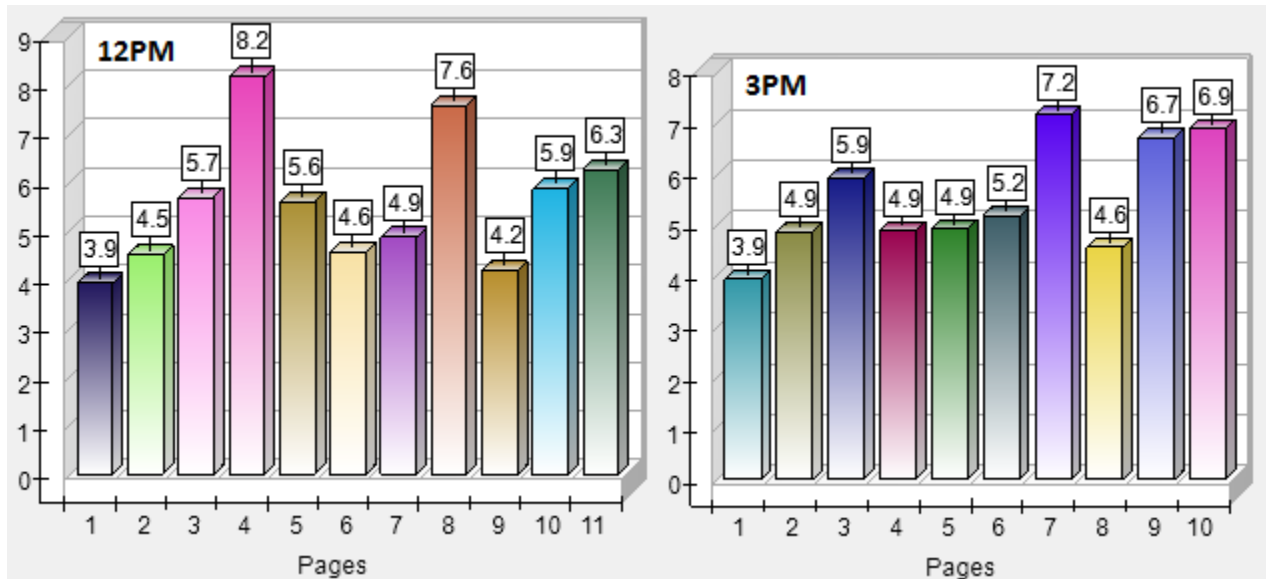


Figure 5.58: Speed variation for CS225 lectures on March 14, 2005

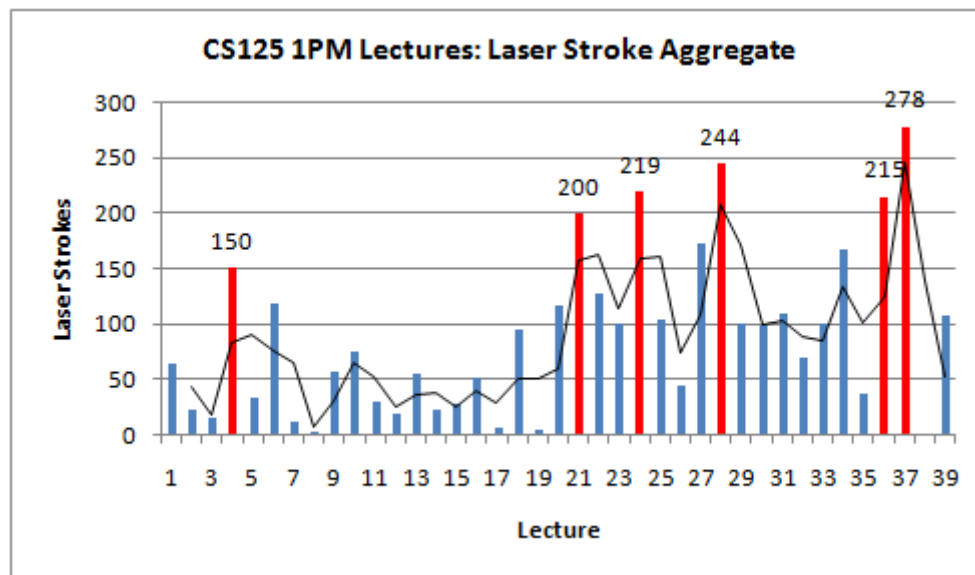
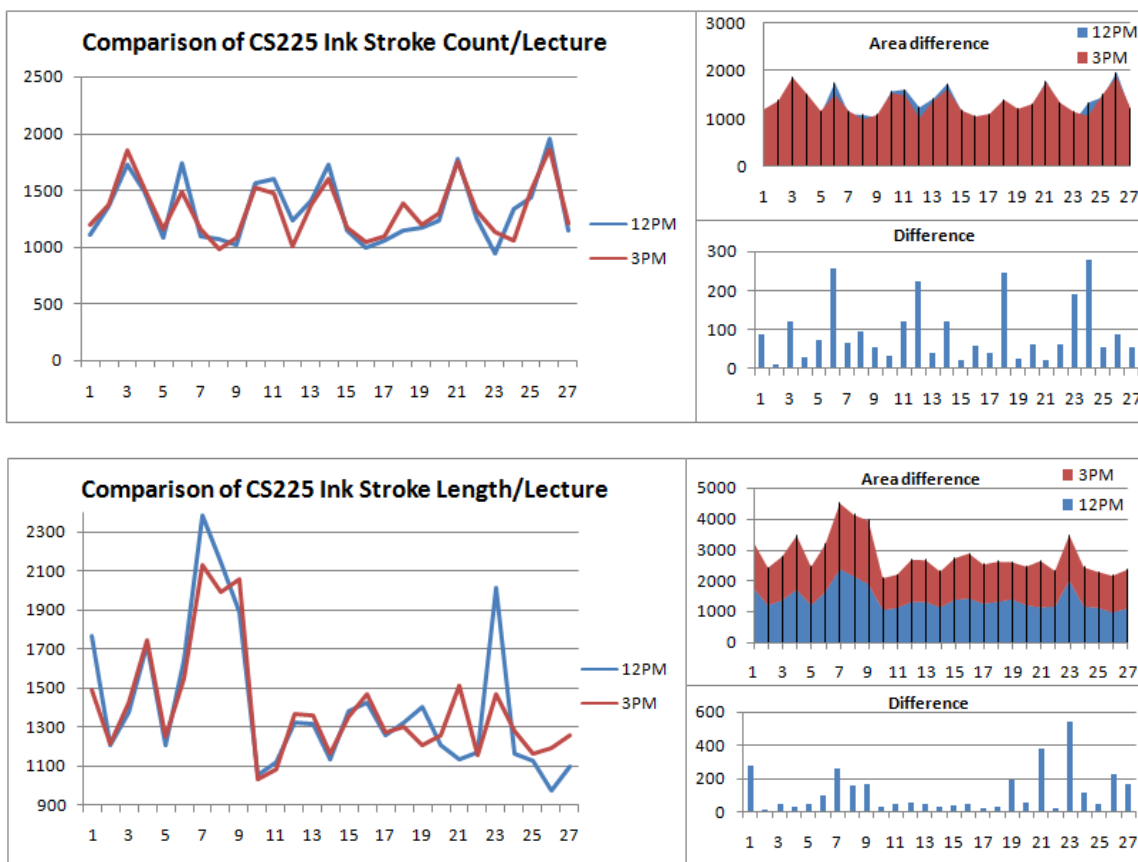
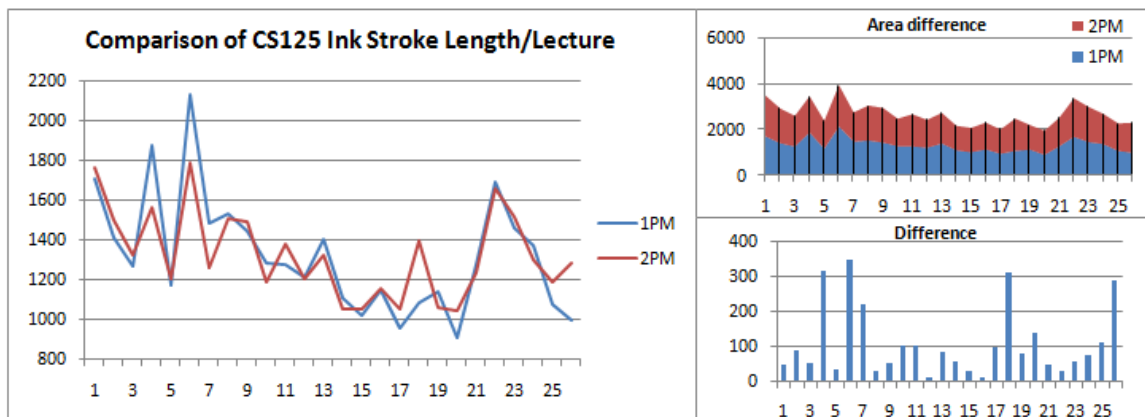
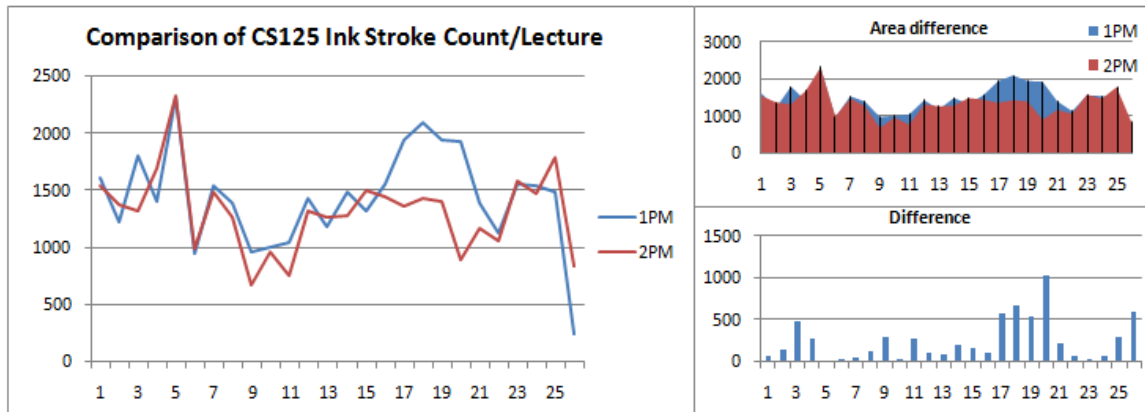


Figure 5.59: Spring 2005 CS125 1PM lecture measurement of the average laser tool usage per lecture. This graph shows that on average, the instructor increased their use of the laser pointer tool towards the end of the semester.

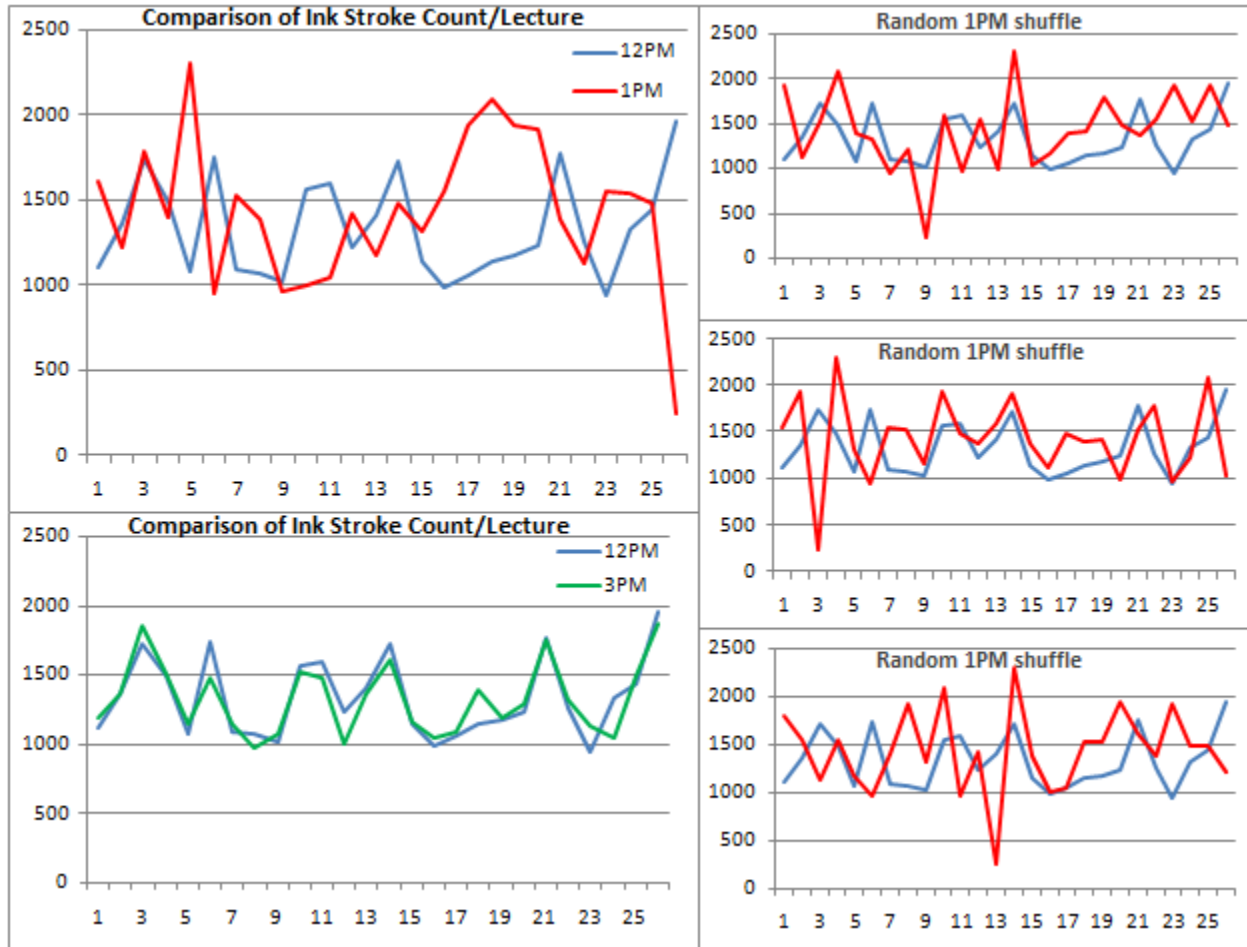
before the first ink strokes on that page, and the time between a subsequent pause in inking behavior. Does an instructor insert a blank slide and immediately write the objective for that slide content? Why does an instructor insert a new slide and not write anything on the new slide? Why not leave the previous slide displayed for students who might still be copying or perhaps still trying to understand/process the previous slide content?

The following four graphs illustrate the similarity (over the duration of the semester) between the two sections of CS125 and CS225 and their respective ink stroke count and average ink stroke length per lecture. Each illustration contains an area chart and bar chart depicting the differences in stroke count between the sections. The sections of CS125 and CS225 show consistent measurements for their ink stroke count and average ink stroke length.





As a contrast to the previous four examples, the graph below presents five comparison of the 12PM CS225 ink stroke count: with the 1PM section of CS125, three random shuffles of the 1PM ink stroke count, and the 3PM CS225 section. Note that the pairing of the 1PM CS125 ink stroke count looks similar to the three random shuffles of the 1PM ink stroke counts. However, the 3PM CS225 section values are nearly identical to the 12PM CS225 section.



5.5.5 Level of granularity of analysis

The previous two sections presented data results based on an analysis performed at the lecture level. Each of the measured attributes reported was calculated as an aggregate value for each lecture. Each lecture's aggregate values were determined by calculating the mean value of each attribute at the page level and taking the mean value for all the pages. One must be careful not to limit an analysis to the granularity of the page since blank pages (as one example) can skew the data results. Another way we looked at the data was at the **ink stroke level**. We enumerated all the ink strokes created in each lecture for the duration of the semester. By abstracting out the pages for the 1PM lecture we had 53,268 total ink strokes. By examining the data at the level of the ink stroke we could scan for patterns that might span multiple pages rather than limiting ourselves to page-level observations. One result of looking at the raw ink strokes in chronological order was a correlation found between increases in speed, length, and pressure of ink strokes.

In searching for characteristics of ink strokes created shortly before a new page was inserted, it turned

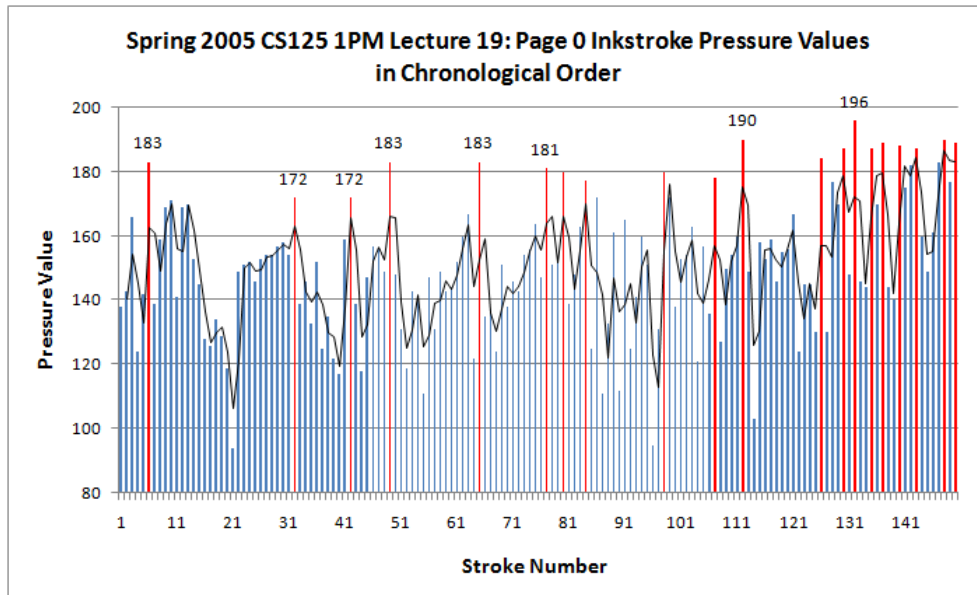


Figure 5.60: This graph shows ink stroke pressure values (in chronological order) for the first page of lecture 19 for CS125. The graph clearly shows a higher frequency of greater pressure values for strokes nearing the end of the list (at about stroke number 111).

out that in a number of lectures the instructor increased pressure (figure 5.60), speed (figure 5.61), and length (figure 5.62) of strokes. It is not clear whether this is a measure of tiredness or simply being hurried.

5.6 Conclusion

In this chapter we presented how two instructors in the Department of Computer Science integrated the use of a Tablet PC whiteboard application into their lectures. Both instructors are extremely talented and successful teachers, yet they have distinct styles. An examination of Jason Zych’s whiteboard application log files for the Spring 2005 semester revealed a meticulous organizer of lecture content who modified his already-effective teaching style to leverage the capabilities offered by the technology. In so doing, Jason was able to provide students with a more detailed record of the teaching process for later review. Students who attended lecture could better recollect what happened in the lecture by reviewing the post-lecture notes in PDF format. A detailed analysis of Jason’s log files illustrates the potential that teaching with a Tablet PCs provides such as the ability to characterize, investigate, and improve teaching practices.

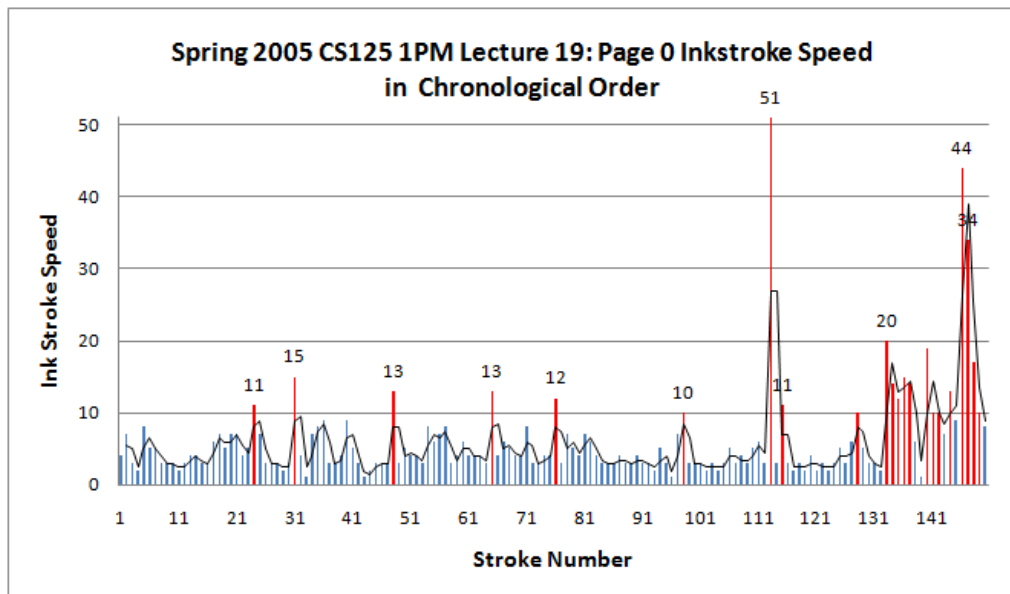


Figure 5.61: This graph provides a summary of the ink stroke speed (in chronological order) for the first page of lecture 19 for CS125. The graph clearly shows a speeding up of ink stroke speed for the last 20 or so strokes.

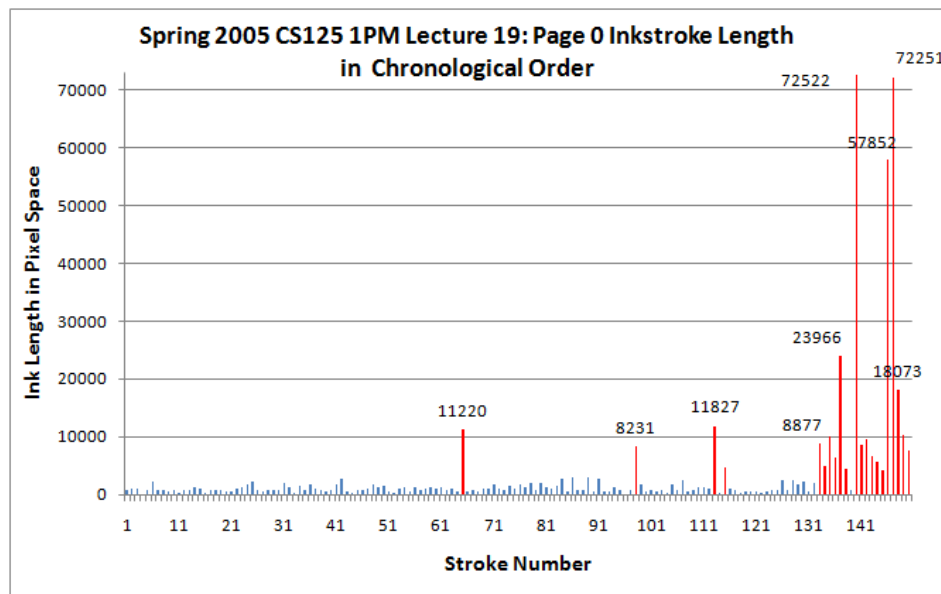


Figure 5.62: Length of all ink strokes for a single page

Chapter 6

Spring 2007 Pilot Study – Supporting Self-paced Instruction

6.1 Introduction

In this chapter we present the design and implementation of a system where a class structure in the form of lecture objectives and assessments was used to provide useful feedback to instructor and students. The term “class structure” refers to the necessary lesson plan that an instructor was required to prepare ahead of time. One of the motivations for building the system was the realization that some students, although they made the effort to attend classes, were not engaged in the lecture. A second motive was the need to provide a concrete context to describe how the students interacted with the Tablet PC devices. Rather than provide a blank slate whiteboard similar to the studies presented in chapter 5, we wanted to keep tabs on student progress (which required planned tasks) in addition to monitoring how well the students were getting along within the structure of the lesson plan. While the requirement of a well defined class structure (i.e. a type of lesson plan) might seem too “strict” a form, the system actually facilitated a more flexible and student-centered learning environment. Teacher-centered instruction became teacher-facilitated learning, with students setting their own learning pace to a great extent.

In section 6.2 we provide an overview of how assessments and objectives were integrated into our system to support a classroom structure (section 6.2.1) conducive to monitoring student progress within the context of the lesson plan. Each objective defined in the lesson plan was designed to work in four stages. The first stage for each objective is the *pre-flight* and is defined in section 6.2.2. The next two stages that follow are *content presentation* (section 6.2.3) and *assessments* (section 6.2.4). The assessment phase is where students spend most of their time while the content presentation phase (where students interact with the instructor to learn an objective) is not a required stage. The final (and not necessarily required) stage is called *post-flight* and is defined in section 6.2.5. In section 6.2.5, we explain how student interactions with their Tablet PC devices were recorded and used by the instructor in a typical lecture. Section 6.3 contrasts the “implicit” feedback collected by the system with an explanation of how explicit feedback was used for students interacting with the instructor. Section 6.4 provides a summary of student evaluations.



Figure 6.1: The SLICE educational technologies laboratory room was setup to facilitate the acronym itself: Students Learn in Collaborative Environments

In section 6.5 we provide an overview of the system implementation detailing the system architecture, the wireless networking challenges, and the solutions we employed, and we conclude the chapter in section 6.6.

In support of the study we changed the layout of the room to be more conducive to collaborative learning (figure 6.1).

As a Lecturer in the Department of Computer Science there were two experiences that led me to completely rethink how I wanted to teach with technology. The first incident occurred in the spring 2002 pilot study (section 4.2.4); once, upon calling on a student to answer a question she responded with “Please don’t ever call on me again, I want to return to being a spectator in your class.” The second experience was a more recent ad hoc experiment we performed in the fall 2006 semester. I was teaching two large lectures (150 and 100 plus students respectively) of CS125–Introduction to Computer Science. After pausing for questions to the class we decided to verify our perception of the classes’ level of engagement. By a show of hands, we asked students to answer the following questions: who is bored, who is interested, and who is lost? We came to realize that at any given moment approximately 30% of the students seemed to be uninterested or off task. To further verify this hunch, we performed a written version of the experiment where students were asked to answer these questions with more specific comments on the in-class worksheet (figure 6.2).

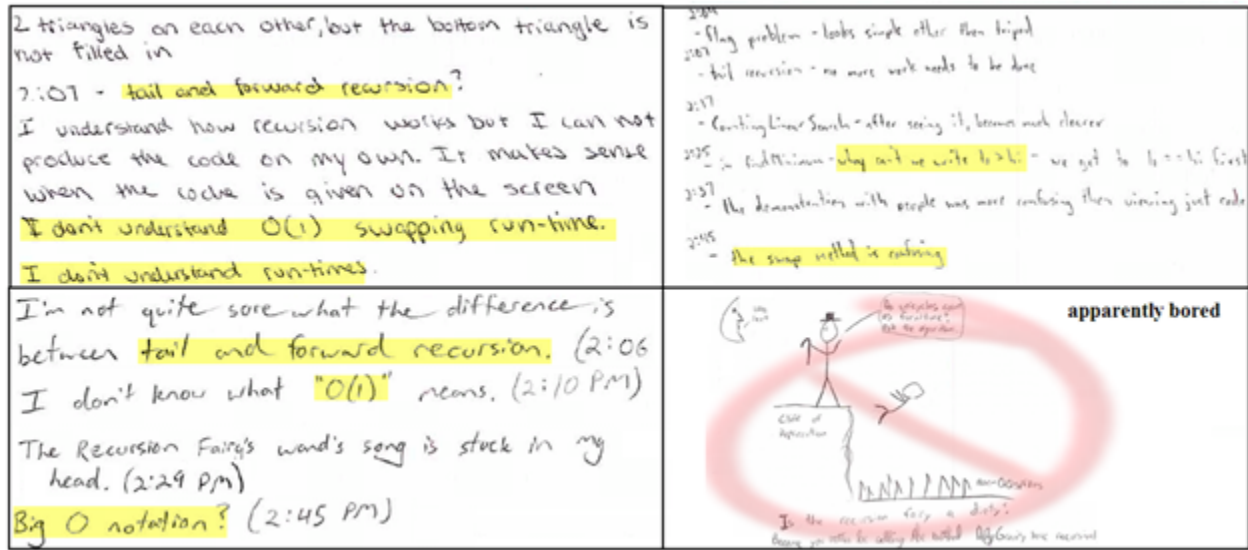


Figure 6.2: The four quadrants represent four student excerpts in the fall 2006 in-class experiment where we asked students to jot down any comments they had at specific synchronization points in the lecture. The highlighted entries are expressions of student doubt. The student in the lower right hand corner is apparently bored and doodled. In a room of 150 students and no technology, instructors are required to “read” the class in an attempt to accommodate the masses. In the spring 2007 study, we would be able to tightly monitor the level of student interaction at much greater level of granularities—at the level of the ink stroke (content) and temporally—whenever we wanted.

6.2 Assessment and Objectives

This pilot study provides an example of how using Tablet PCs in the classroom can allow for fundamentally new classroom structures. In the spring 2007 semester we taught an experimental section of our introductory programming course (CS125) and repeated the study in the summer semester for a Data Structures and Software Principles course (CS225). Of particular interest to us was to determine how the student’s performance and classroom engagement was affected by the level of their prior knowledge of the subject matter.

6.2.1 Classroom structures

Classroom structures refer to the different modes of interaction possible between instructor and students in a learning environment [4, 94]. For example, the scenario where all students are working independent of one another, on the same task, under the close supervision of the instructor, is one such structure. In contrast, consider a scenario at the other end of the spectrum where students are performing independent tasks without the supervision of the instructor. The use of the machines allows us to support structures that were not possible in the past. Assuming that students were on task, we wanted to ask the questions: What

is each person doing at any given time in the classroom, and how is it being controlled? If we could answer these questions with some certainty then we would have implemented a reasonable classroom interactional structure.

By integrating planned-out assessments, our system “knew” what the students were doing at any given time. In this chapter we describe an example classroom structure whereby students were either locked into an instructor’s control, free to perform formalized (prepared ahead of time) assessments, or part of a synchronous group of students paying attention to the instructor. The instructor’s ability to keep tabs on the various student groups was made possible by the software which helped the instructor to maintain awareness in the multitasking environment. To use this tool properly, teachers will have to have their classroom structures planned ahead of time (hence the “strictness” of the classroom structure).

Designing different classroom structures, we consider how the level of task management and measure of similarity of activities among students in the classroom could affect the learning environment. Figure 6.3 spatially models representations of classroom structures in a matrix. The data along the Y-axis represents the level of task management required by an instructor and the data along the X-axis represents the level of independent student activity. In the figure, point A represents a classroom structure where the instructor is in control of the classroom and all of the students are performing the same activity. Point B represents a middle ground where an instructor begins to undertake multiple tasks while students are working independently. Point C represents a classroom where an instructor is trying to do several tasks and the students are doing different activities.

The structure we experimented with was an attempt to bridge the gap between students who typically fall behind in lecture and those who are quickly bored. It is a partially self-paced class, with occasional “synchronization” points. The teacher plans each lecture as a sequence of objectives and each objective contains the following components:

- Pre-flight (section 6.2.2) – a brief question which summarizes the topic to be taught. Students submit a “yes” or “no” response indicating that they can or cannot answer the question.
- Content presentation (section 6.2.3) – the instructor teaches the topic represented by the objective.
- Self-assessments (section 6.2.4) – students have the opportunity to perform assessments to test and verify their knowledge of an objective.
- Post-flight (section 6.2.5) – questions performed synchronously as a class. They serve as a recap and reinforcement of material learned. The main purpose of the post-flight is to provide students with feedback on their understanding of the material and to send these results to the instructor.

The components required for a given class session were described in an XML file (for all intents and purposes the “lesson plan”) detailing the objectives to be covered in the lecture, and a PDF document for

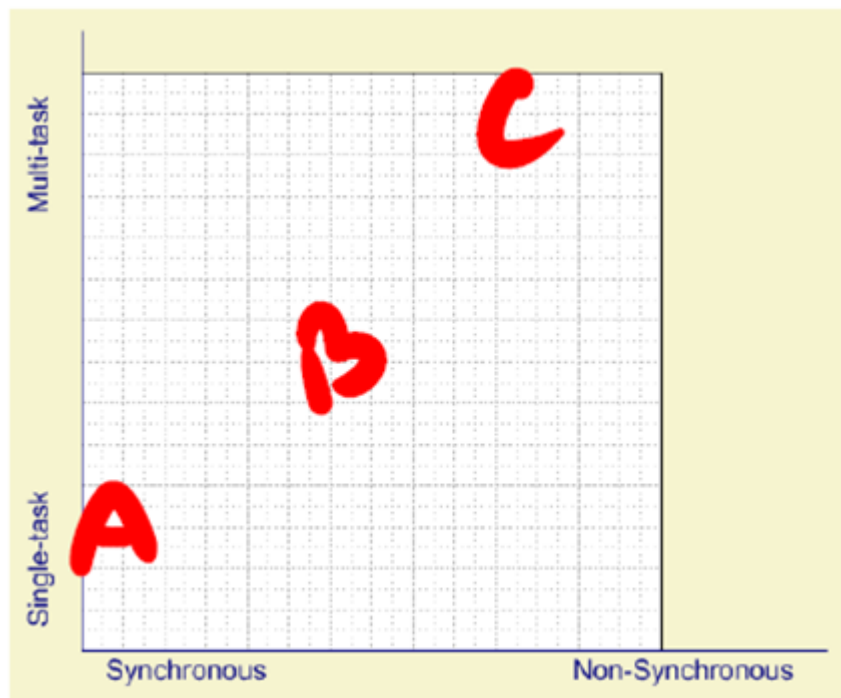


Figure 6.3: This matrix is an attempt to spatially model representations of classroom structures

each objective listed in the XML schema. For example if the only objective for a lesson plan was to introduce the properties of a binary search tree, our XML file might include the following entry:

```
<objective name="Properties of a Binary Search Tree" file="bst.pdf"/>
```

All PDF files listed in the objectives schema require at least five pages to be considered valid “objective” documents. The first page of the PDF document represented a pre-flight (section 6.2.2) assessment, followed by n pages representing <assessments, solutions> where each assessment page contains a number of questions and the subsequent page includes both the assessments and their solutions. The last two pages of each PDF document always concluded with a single post-flight assessment (a question at a higher level of difficulty that a student should expect to be able to solve on an examination) and its solution. Figure 6.4 illustrates the PDF document just described and the complete document can be seen in appendix D.3. These PDF documents did not include actual lecture material.

There are five steps required to set up and run a typical class session:

1. A lecture begins by first creating a Sharecast® *session* (the Sharecast network protocol is discussed in section 6.5.5).

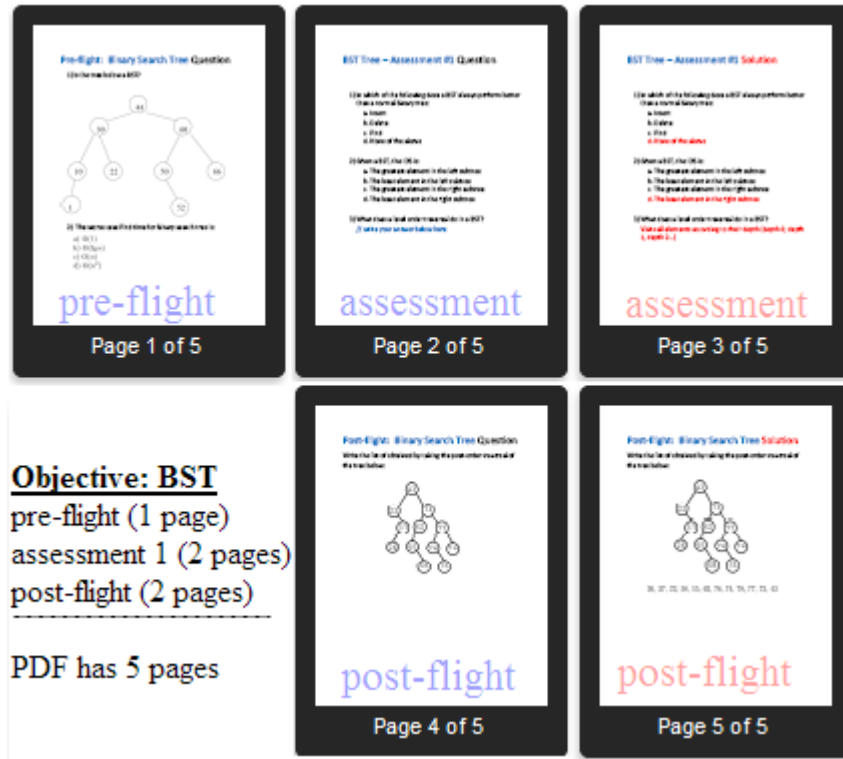
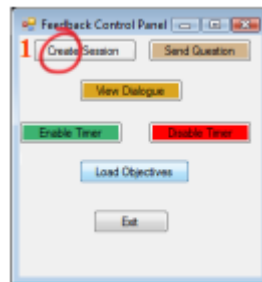
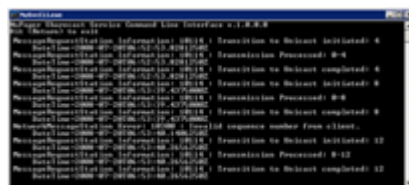


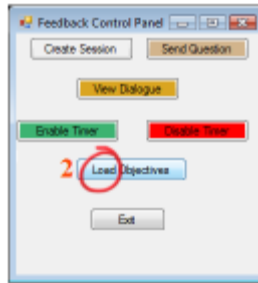
Figure 6.4: The contents of an example PDF document for one objective of a lesson



which results in the addition of the network console screen



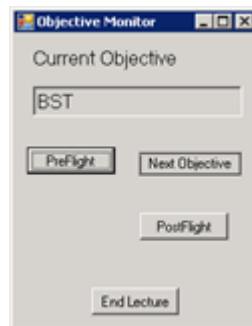
2. The instructor then loads the objectives for the lesson:



by loading the XML file containing the objective names and associated PDF files:

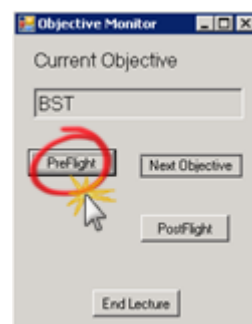
```
<?xml version="1.0"?>
<objectives>
<objective name="BST" file="bst.pdf"/>
<objective name="Insertion Algorithm" file="bstInsert.pdf"/>
<objective name="Remove Algorithm" file="bstRemove.pdf"/>
</objectives>
```

which results in two additional windows: the objective monitor



and the teacher's display

- The instructor then initiates the first pre-flight (section 6.2.2) of the session using the objective monitor:



at which time all students perform the pre-flight:



4. The student responses (either “yes” or “no”) are tallied and sent to the teacher’s display:

Objective	Number Attempting	Users
BST	14	phatw2, gphw2,
Assessment 1		
Assessment Algorithm		
Assessment 1		

Objective	pre-flight		Postflight	
	Yes	No	Yes	No
BST	10	4	0	0
Assessment Algorithm	0	0	0	0
Assessment 1	0	0	0	0

Students who submitted a response of “no” to the pre-flight (four students) are automatically switched into note-taking mode and await the instructor’s lesson on the first objective:

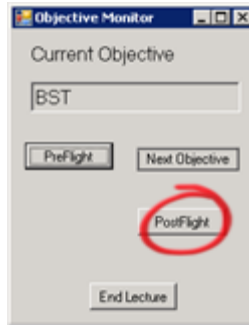


Students who responded “yes” (ten students) remain in assessment-mode and begin taking assessments for the current objective:

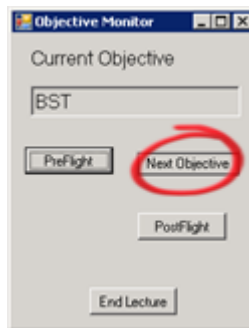


5. After the instructor has completed the lesson for the current objective she can either:

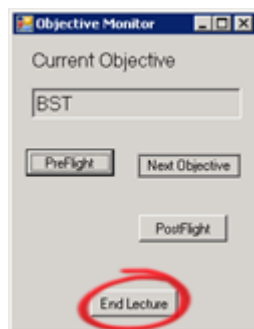
- perform a post-flight for the current objective:



- or move on to the next objective by selecting *next objective*:



- or end the lecture by selecting *end lecture*



in the objective monitor.

An instructor can end the class session at any time. The result is the creation of a log file for all student interactions for the class period (see appendix E.2). Students are given the control to logout of the system by clicking “finish” at which time the contents of their class session are saved into a PDF formatted file and uploaded to a web server. Figure 6.5 is a graphical representation of the classroom structure.

For the most part, the class goes through the lesson objectives together, but may move at different rates within an objective. Students completing all but the final assessment for an objective (the post-flight) are free to move on to subsequent objectives prior to the synchronous post-flight of previous objectives. Very often, the instructor will request that these students work with the students who had not yet completed the objective; the “teacher’s dashboard,” described below, makes this possible.

The synchronous aspect of the system allows an instructor to force every student in the class to the post-flight page in their documents. Once this occurs, students are no longer able to traverse back to previous

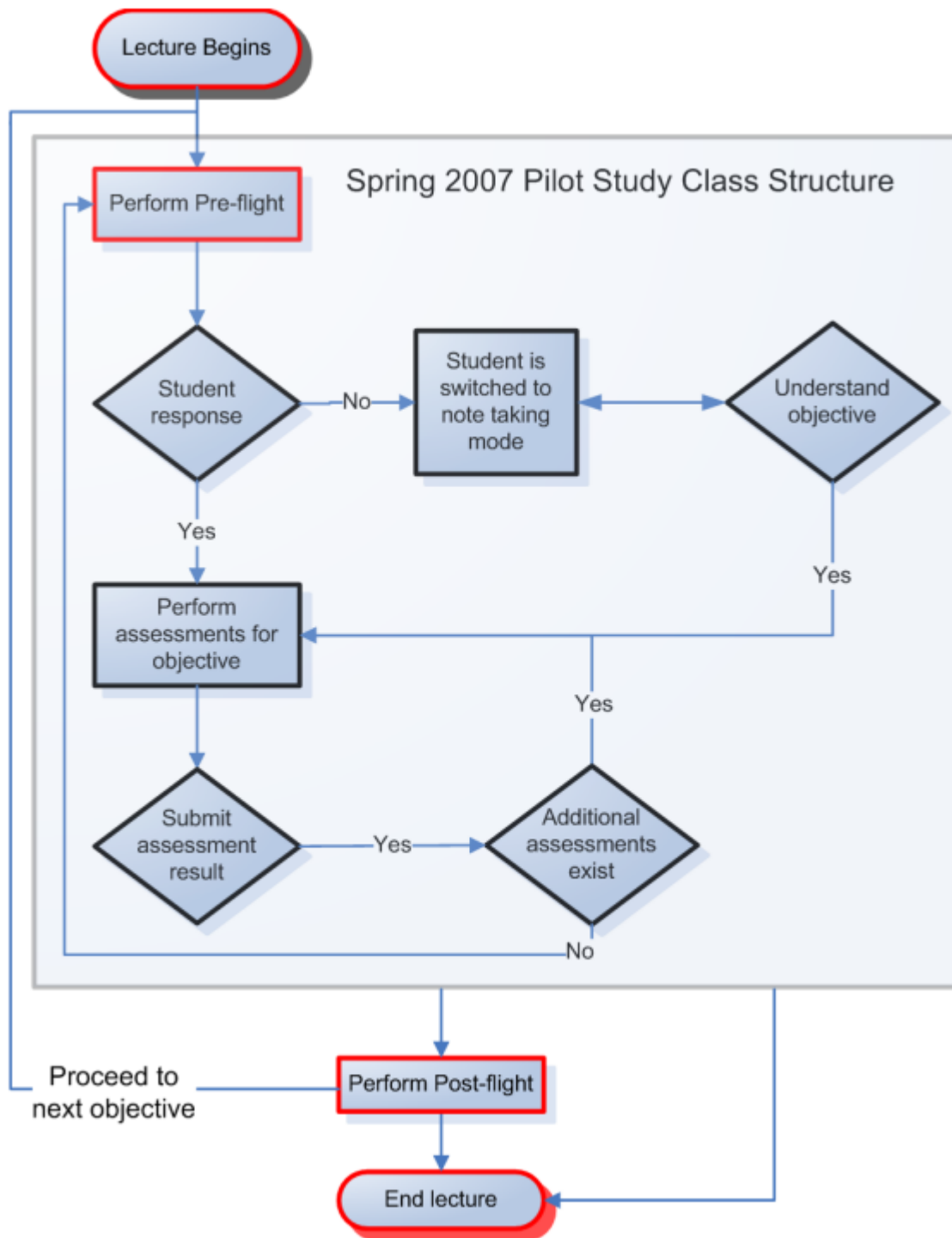


Figure 6.5: This flow chart illustrates the pattern of instruction (flow of interaction) for the spring and summer 2007 pilot studies. The entities outlined in red (Lecture Begins, Perform Pre-flight, Perform Post-flight, and End lecture) represent processes that can be synchronously forced by the instructor's objective control panel.

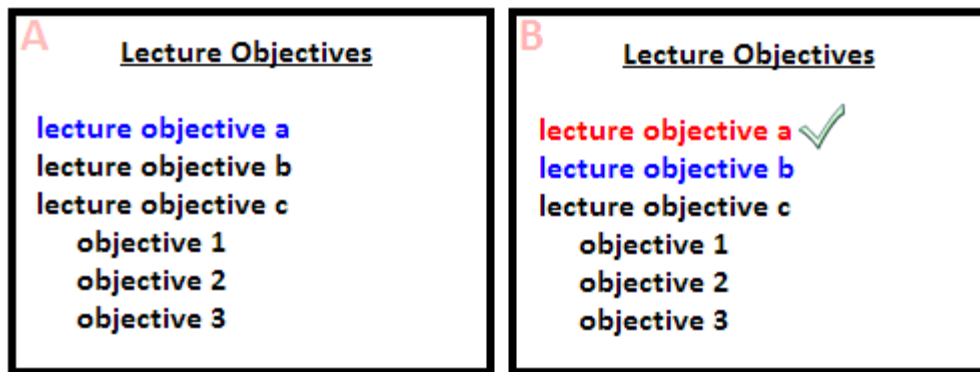


Figure 6.6: Example of two states of the dialogue box displaying the status of lecture objectives displayed on the projector. A) Objectives in blue represent the current objective B) The **RED** color of a lecture objective and check mark indicate that an objective has been completed.

assessment pages. By providing a display of the current and subsequent lecture objectives (figure 6.6), students are more likely to subscribe to instructor discussions of interest.

In addition, this structure provides a mapping of lecture objectives to student created notes while also providing a roadmap of the lecture content. Overall, objective mapping provides a recordable context of what and when an objective was taught in the classroom session (figure 6.7). For example, at any moment in the lecture a student can determine what the objectives for the lesson are, what the current status is (the current objective being taught) and when an instructor moves on to subsequent objectives.

At any point in the lecture a student is in one of three modes: assessment mode, note-taking mode, or “outside” the application. While the students are in note-taking mode the system provides an association between what they are writing and the current instructor objective. This association facilitates searching for specific course topics in the student created documents. The class also incorporates a window on the teacher’s computer—which we call the “teacher’s dashboard” (figure 6.8)—indicating how many students are at each point in the current objective. This is an illustration of how a carefully designed environment allows for the exploration of different pedagogies. We will revisit this interface in section 6.5 when we describe the details of the implementation and classroom integration of our system.

6.2.2 Pre-flight

Historically the Physics department has pioneered learning technologies [201] and scored at the top of the heap on ICES¹ large lectures. The fall 2007 Physics 101 course syllabus describes the importance of pre-lecture readings and pre-flight activities (figure 6.9). The course has historically incorporated the use of

¹The acronym ICES stands for Instructor and Course Evaluation System and was developed at UIUC as a way to collect information from students about courses and teaching (<http://www.cte.uiuc.edu/dme/ices/index.htm>)

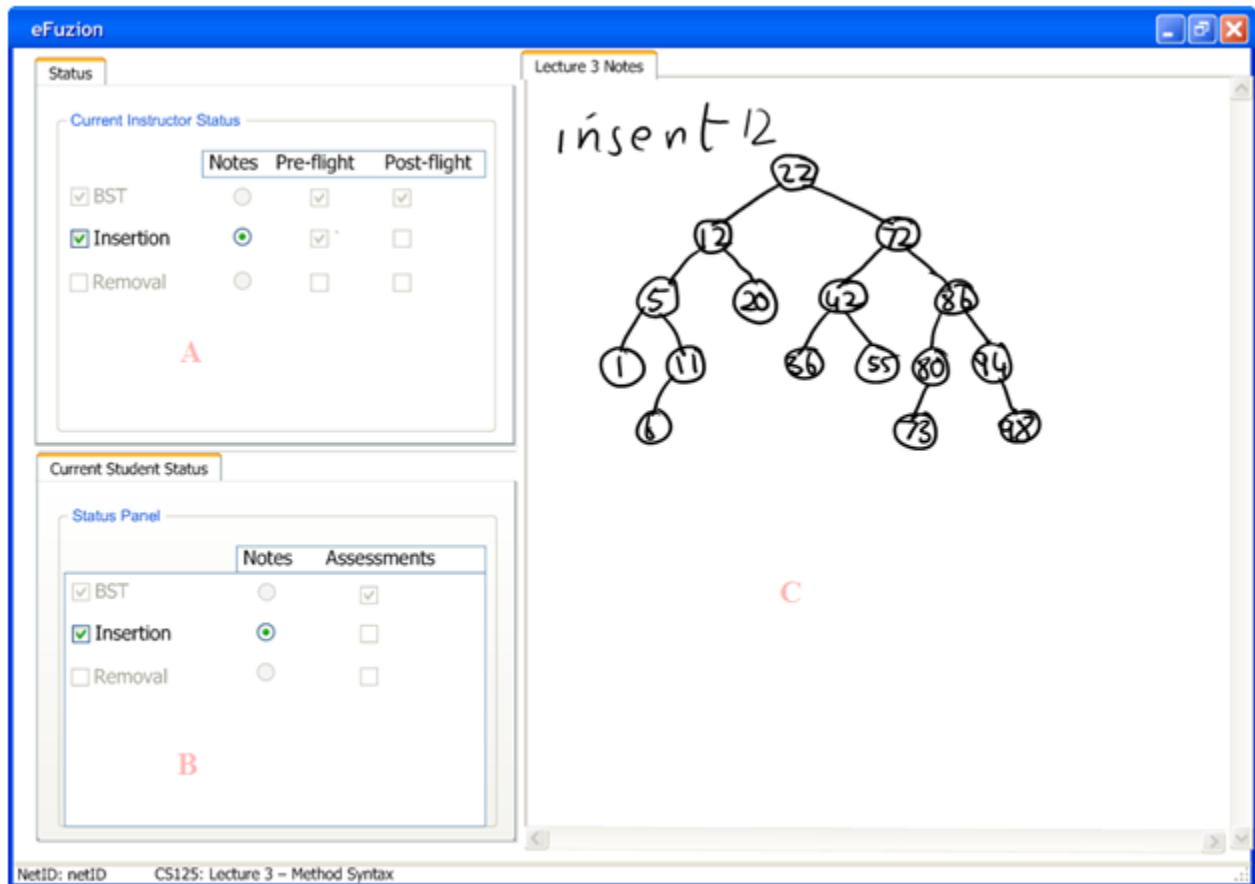


Figure 6.7: The first design of our student user interface. A) The instructor’s current status shows that the first objective (BST) Binary Search Trees has been completed including the pre and post-flights. The current objective is the Insertion algorithm and the pre-flight has already completed. B) The student authoring this document (Lecture 3 Notes) is currently in **note-taking** mode. C) This represents the student’s whiteboard space (note-taking pane). Any ink added to this panel is automatically associated with the “Insertion” objective since the instructor is currently teaching that objective. The association is added as meta data to the document.

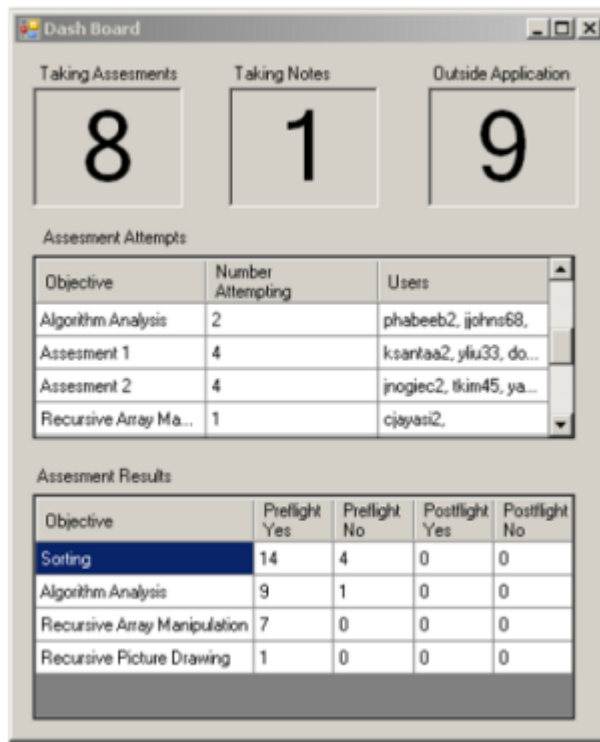


Figure 6.8: Dashboard displaying the current status of students at the conclusion of a lecture. At the time of this screenshot, eight students were taking assessments and nine students were outside the scope of the application. The current objective was Algorithm Analysis, and nine students were able to answer the preflight as illustrated by the number ‘9’ in the Preflight Yes column under Assessment Results in the dashboard

Physics 101 Lectures

Spring 2008

Before 6am on the day of each lecture you must complete the PreLecture and Preflight.
The lecture handouts should be brought to each lecture.
After each lecture, the complete lecture slides will be available.

Figure 6.9: Physics 101 course syllabus at the University of Illinois at Urbana-Champaign

pre-lecture and pre-flight activities as a means of preparing students for the lecture.

Pre-flight design

The pre-flight in our system was designed to provide students with the responsibility of quickly assessing how they perceive their understanding of a given course objective (topic). In his New Taxonomy of Educational Objectives [124] Robert Marzano introduces the *self-system* as one component of his overall taxonomy. The self-system is comprised of the attitudes, beliefs and feelings that determine an individual's motivation to complete a task and include *importance*, *efficacy*, and *emotions*. When a student is confronted with a learning activity, one of their initial responses is to determine how important the task is to them. Is the task something they want to learn or believe they need to learn? Will expending the energy to pay attention and participate in lecture help them to accomplish this goal? Students who are uninterested hinder their own learning. In our experience, some students decide to spend class time working on machine problems (programming assignments for the course) instead of what the instructor has requested (i.e. some type of learning assessment). As a result we built into our system the ability to measure a student's level of interest using explicit polls (section 6.3) and to essentially force students to stay on task by pushing out explicit assessments to their devices.

One concern we had in designing our system was figuring out what reasonable assumptions we could make about student participation. Requiring pre-lecture readings is always a good idea. We assume that college level students possess the maturity and motivation to learn and prepare for lecture. Unfortunately, students are often misguided by the principle of EDF—Earliest (graded) Deadline First. If we do not assign it, the students will not do it. And, if they are going to complete the assignment, they will certainly procrastinate for as long as possible.

A second concern was attendance. When we began piloting our CS1 course, we saw declines in student attendance just before the midterm examination period. (This was not a problem for the graduate students enrolled in our Data Structures and Software Principles for Non-Majors course in spring 2002, described in chapter 4, section 4.2.4 In fact, attendance there was consistently at about 95%, most likely because these

were mature students with a deep desire to learn.) Given the restrictions on the number of credit hours our course required, and the fact that the semester had already begun, we were not able to alter the syllabus to require students to perform assessments on a consistent basis outside of class. We would need to resort to the use of on-the-fly pre-flights to quickly gauge what we hoped students had already been exposed to before the lecture.

Pre-flight implementation

The pre-flight was the starting point for all objectives. Before an instructor began teaching a topic students were required to take a pre-flight, in the form of a quick poll. The results of the poll provided the instructor with a quick aggregate of how many students “believed” they had a firm understanding of the objective. The pre-flight assessment was not a rigorous exercise but rather a quick assessment to find out if a student already understood the topic. That is to say, it measures a student’s perception of whether they understood an objective on the teacher’s agenda. See [171] for an example of large-scale deployment of aggregations of student in-class responses using the Classroom Learning Project (section 2.8.1). We considered a few graphical user interface designs for the pre-flight input dialogue box such as the one illustrated in figure 6.10. The instructor could initiate a pre-flight by clicking on the pre-flight button in the objective monitor. At that time, any student who had **not yet** submitted a pre-flight for the current objective was forced to do so (figure 6.11).

6.2.3 Content presentation

Content presentation referred to the mode of interaction where the instructor presented mini-lectures of about ten minutes duration. Students who answered “no” to the current pre-flight were automatically “subscribed” to participate in the content presentation. That is, the application automatically switched these students from assessment mode into note-taking mode. The teacher’s display (section 6.3.1) provided the instructor with a control panel showing the current state of the class: the number of students in note-taking mode, the number of students in assessment mode, and the number of students outside the scope of the application (in all probability off task).

Presumably it was possible for a student to answer “yes” to a pre-flight and upon attempting the first or subsequent assessments realize they were not as familiar with the material as they thought. Under these circumstances, if a student switched from assessment-mode to note-taking mode, their change of status was recorded by the system and pushed to the teacher’s display. Because it can be difficult for a student to jump in to the middle of a lecture discussion, the change of status for that student became a way to communicate to

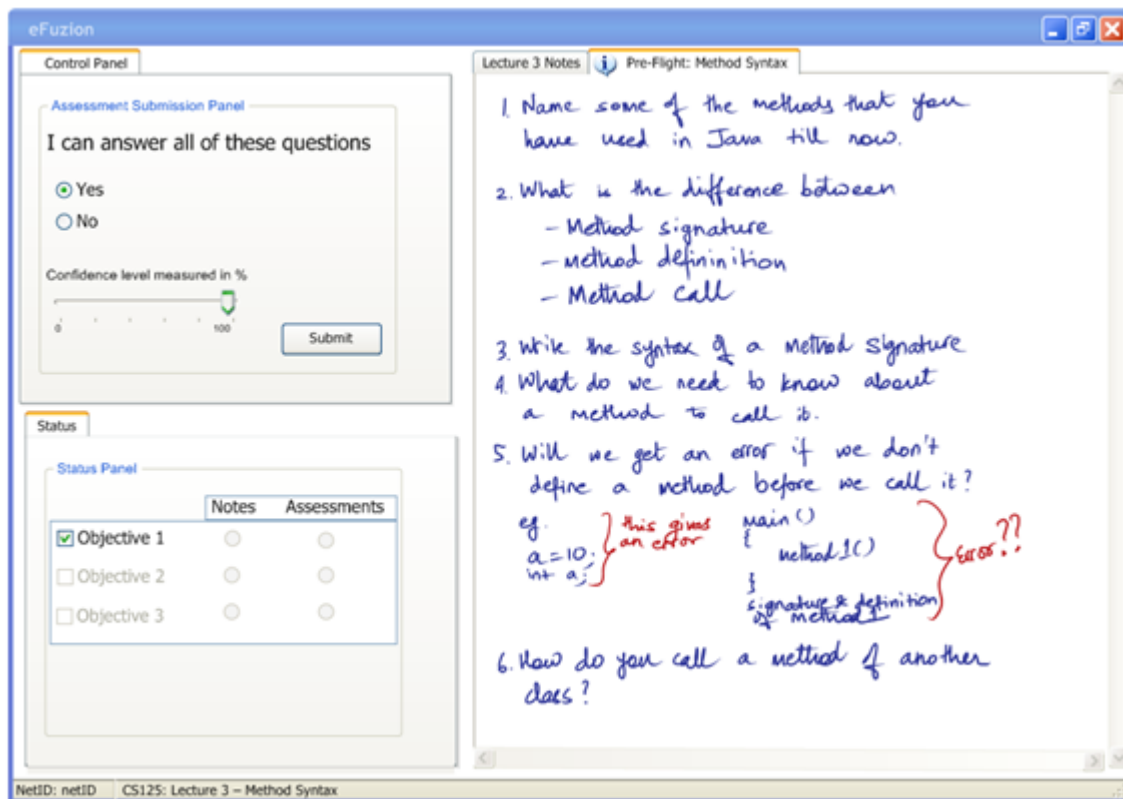


Figure 6.10: First design of a graphical user interface for the pre-flight in our system. The whiteboard contains the pre-flight question. The student's task was to read the content of the question and answer whether they understood **all** of the content or not. One of the design decisions we needed to make was how to generalize the mechanism for asking the students for their pre-flight result. What if there was just one question instead of a page of short questions? We would later resort to a simple "yes" or "no" button. The status portion of the figure was meant to provide a feedback mechanism for students as they complete the lesson's objectives and assessments.



Figure 6.11: Objective monitor control allowed the instructor to initiate pre-flights, post-flights, move on to the next objective, and end the lecture. Upon clicking the "end lecture" button all of the statistics that had been accumulating on the server from connected clients was written out to an XML formatted file.

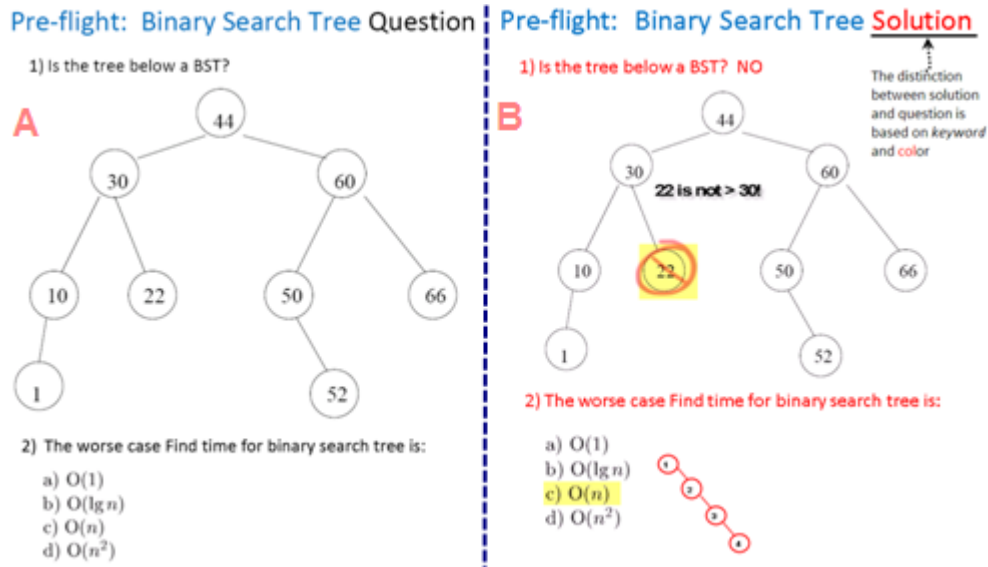


Figure 6.12: A) An example of a pre-flight question on binary search trees. The student submits either a “yes” or “no” response indicating whether they believe they understand the material and can answer the questions. B) The solution to the pre-flight assessment. Only students subscribed to the “current mini-lecture” will see the solution (if the instructor projects the solution) in class. Post lecture, the solutions to the pre-flights were not included in their documents, although the remaining questions and solutions were included in the PDF archive of the lecture.

the instructor that additional students required her attention. The system provided a real-time queue where students explicitly expressed their interest in hearing the mini-lecture when it was next offered. We used the analogy of a bus schedule to illustrate how students could catch the next mini-lecture for an objective in the event a student missed the previous episode. When the queue was empty (no students were in note-taking mode) or an instructor decided it was time to move on, the instructor either initiated the post-flight or proceeded to the next objective. In one implementation of the system, an instructor was not required to perform the post-flight in order to move on the next objective. However, we ended up using the post-flight questions as a means of encouraging students to continue the learning process outside of lecture. Since they were under the impression that being able to solve the post-flight assessments would greatly enhance their performance on the exams, it was an easy sell.

As an example, consider the following pre-flight question on Binary Search Trees on the left-hand side of figure 6.12. The questions were always identified by the keyword “question” in black. The instructor’s mini-lecture on an objective usually began with an explanation of the solution to the pre-flight (also illustrated in figure 6.12). The students who answered “yes” to the pre-flight were not given the solution. The rationale behind this decision was that an honest assessment would not require a verified solution. If there was any doubt in the student’s mind, they should have responded “no” to the pre-flight.

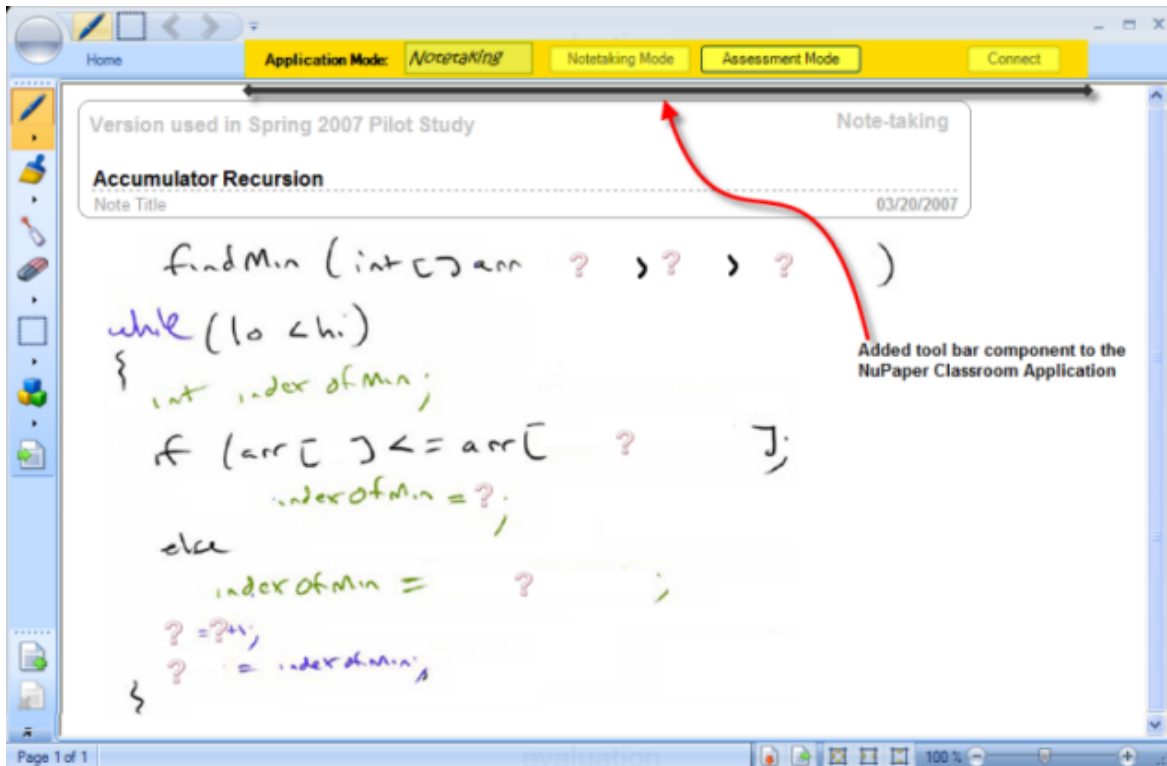


Figure 6.13: User interface for note-taking mode in the spring 2007 pilot study

Recall that the solution to the pre-flight was not provided to the students upon submitting their response. Assessments, however, did provide such immediate feedback (section 6.2.4). In contrast to the pre-flight, after a student submitted their response to an assessment an example solution was immediately pushed to their screen. The motivation for doing this was to help the students to evaluate (to the best of their ability) whether or not they understood the material. The idea was that a student answering ‘yes’ to the pre-flight and failing to provide a solution to an assessment that was anything close to the instructor’s solution was in need of assistance. Upon taking a pre-flight and answering “no” the student was immediately switched to note-taking mode; the instructor was signaled that the student was subscribed to the mini-lecture. Figure 6.13 illustrates the user interface for students in note-taking mode for the spring 2007 pilot study. It would have been prudent for us to minimize the note-taking interface to exclude some of the unnecessary tools such as the shape tool and the laser pointer tool.

For the summer 2007 CS225 pilot study we modified the NuPaper® application to include some of the data structure modules that were available in the e-Fuzion summer 2002 study (chapter 4 section 4.2.2) such as the stack, array, and matrix data structures (figure 6.14). We also included a snapshot tool which allowed users to quickly import anything visible on their screens into the whiteboard application. Two additional modifications were made to the user interface. To facilitate the creation of the data structures

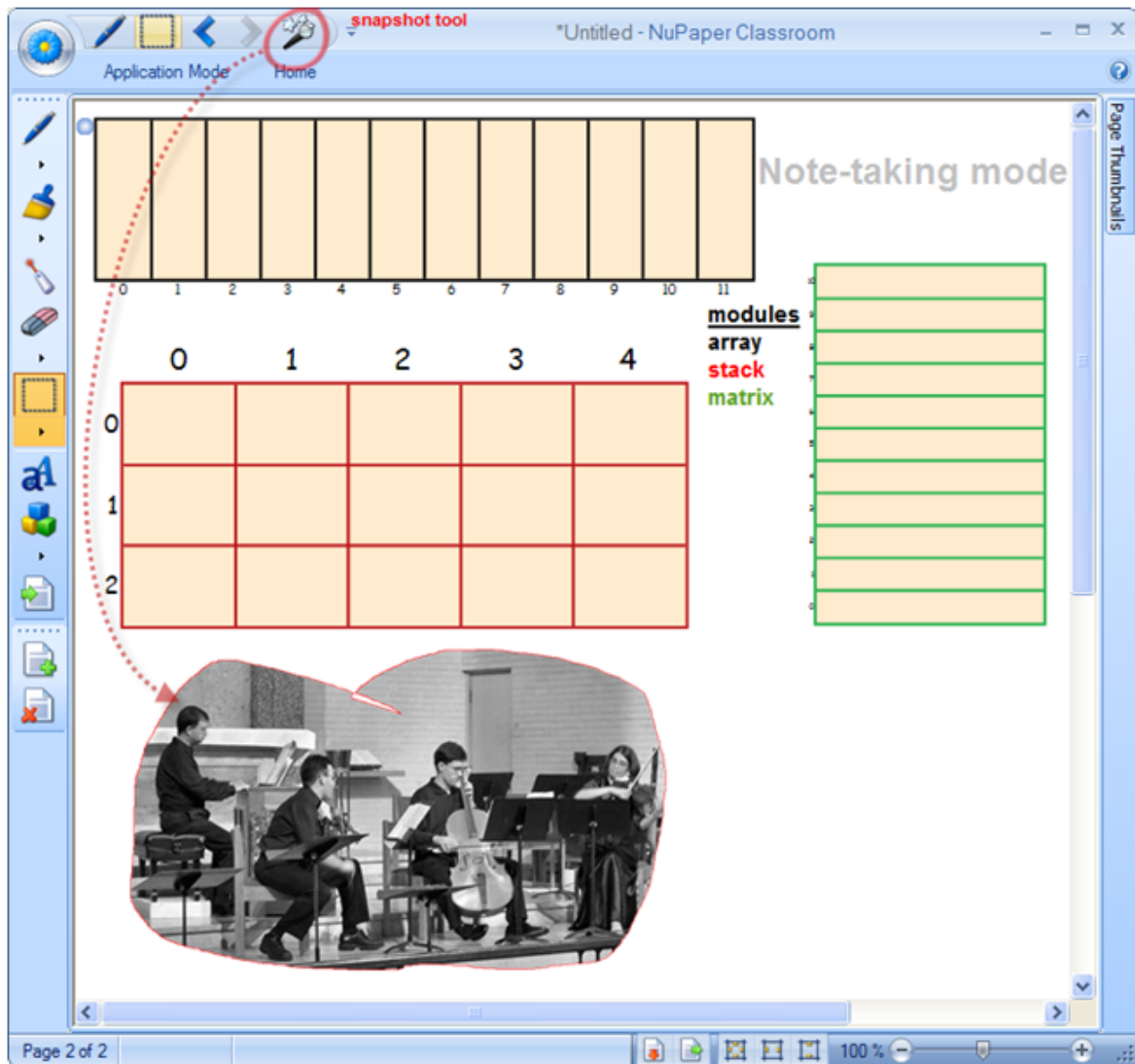


Figure 6.14: Modified interface for the summer 2007 CS225 pilot study

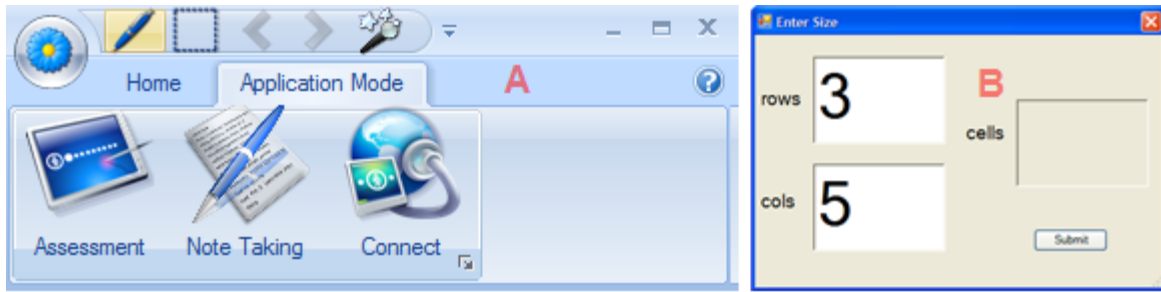


Figure 6.15: Summer 2007 application interface

an ink-based dialogue box was added to support user input for the number of cells or rows and columns (part B of figure 6.15). The ink-based user input was checked for valid integer values and automatically sent as parameters for the inserted modules. The second change was the addition of a ribbon toolbar to allow students to more easily switch between note-taking and assessment modes (part A of figure 6.15).

6.2.4 Assessments

Recall from section 2.2 that an assessment is defined as the process of documenting, usually in some measurable terms, knowledge, skills, attitudes and beliefs. Each objective had associated with it a number of exercises or assessments. The pre-flight and post-flight were not considered “assessments” (although to the student they might have appeared to be so). Students were free to attempt objective assessments at their own pace but were required to attempt assessments in chronological order and could not attempt post-flights on their own. In the event an instructor did not have sufficient lecture time to commence a post-flight, we designed the system so that students would have the post-flights available in their post-lecture documents (along with all of their pre-flight responses, objectives assessments, and instructor example solutions).

Our system supported two types of assessments; forced choice assessments and short written responses. Free choice assessments attempted to measure basic details related to a measurement topic—such as vocabulary terms, facts, and time sequences—straightforward from a learning perspective. These types of assessments are typically used to demonstrate knowledge of details through recognition and recall of generalizations and principles. The integration of these assessment types allowed for a semi-automated evaluation in real-time of student responses which an instructor might choose to send to the dashboard for a summary of student results. These includes multiple choice, matching, alternative choice, true or false, fill in the blank, and multiple response questions.

Short written responses are a type of constructed response. These response types required students to construct a correct answer as opposed to recognizing one. As an example, consider an activity that

requires students to provide self-reflection. Our system supported open-ended assessments that had students articulate their perceptions regarding their learning of objectives. K. Patricia Cross [39] developed some techniques to allow students to reflect on what they learned and what was still confusing to them—such as the minute paper: before the end of a class period, the instructor asked students to write brief answers to two questions: What is the most important thing that you learned in class today? What is the main answered question you leave class with today?

A variation on this exercise is what some refer to as the muddiest point [132]—students simply describe what they are most confused about in class and the teacher uses the information to plan further instruction and organize students into groups [94]. It is important to note that the category of classroom assessments in our system were not the same as polls which are a type of explicit (obtrusive) assessment created and sent out on-the-fly to quickly receive feedback. Section 6.3 presents our implementation of such explicit polling. Figure 6.16 illustrates an example assessment from the summer 2007 semester of CS225 for the objective C++ programming. Note the reduced interface on the left hand side in the assessment mode; students were limited to a pen and erasure tool while in assessment mode. After answering the questions in this assessment the student submitted their response by clicking the “submit” button. They also had the option of specifying a confidence measurement if they chose to do so. The default value was 100.

6.2.5 Post-flight

The design of the post-flight was a bit more complicated. We wanted to provide a way for students to articulate how they would go about answering a more difficult assessment without taking away from lecture time. Initially we did not know what type of feedback would be useful aside from a “yes” or “no” response representing the student’s assessment whether they could or could not answer the question (figure 6.18). This was not just an attempt to place the responsibility of learning on to the students, but also a way to provide a realistic expectation of the level that the instructor expects the students to be able to perform to. In the event that a student was able to complete all the assessments for a given objective, they should have theoretically been able to complete the post-flight exercise. However, since we designed the post-flight exercises to be of equal difficulty to an exam question, more time was required to answer these assessments. Of course, this depended on how well the assessments for a given objective were written. Ideally, they were designed to iterate through the different learning levels such as those illustrated in figure 2.1.

By comparing the results of a student’s pre-flight and post-flight response we could deduce whether the student believed they learned the objective. The challenge was how to design a post-flight where students could quickly engage in a higher level/order of thinking and in a short timespan decide how to proceed. We

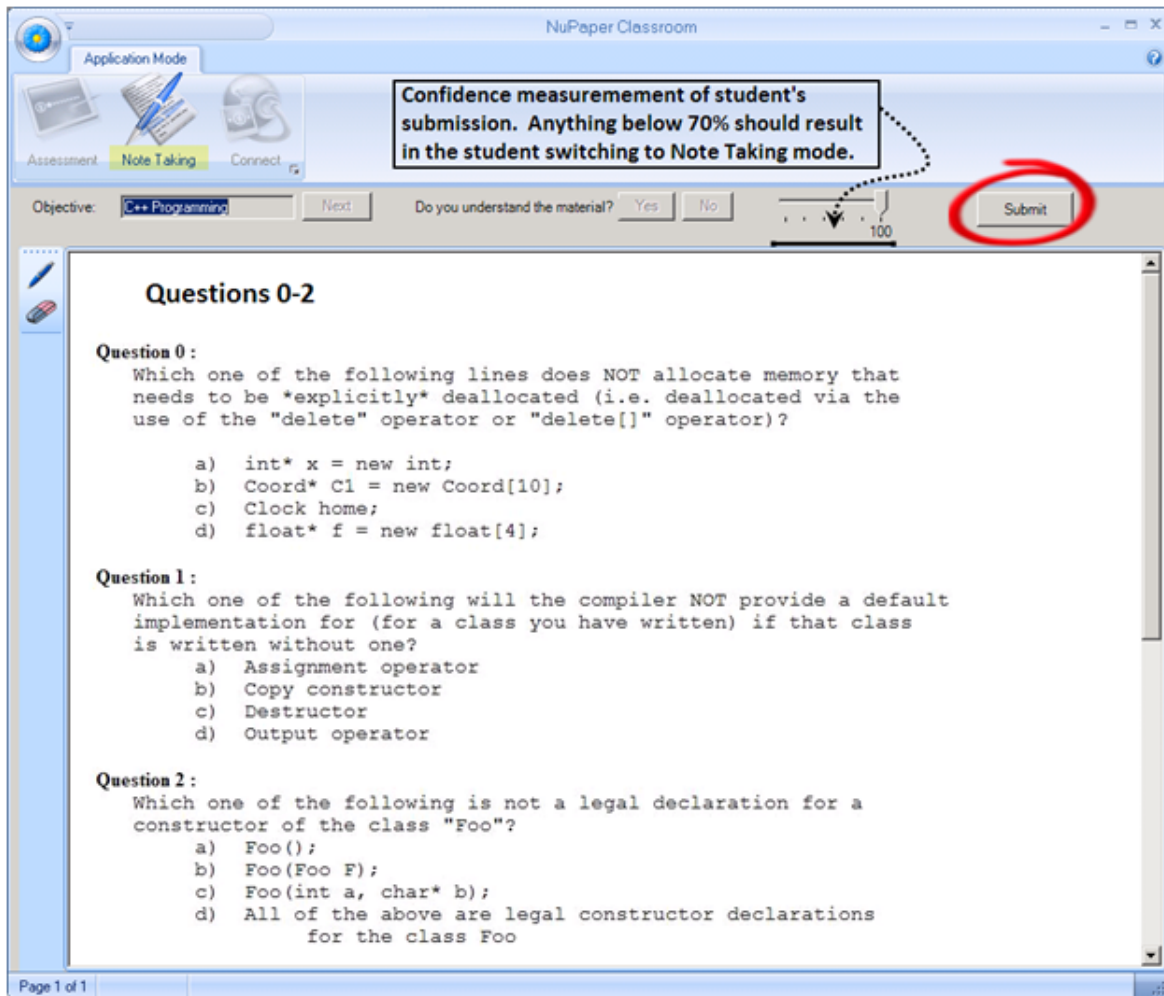


Figure 6.16: Example of an assessment for the C++ programming objective

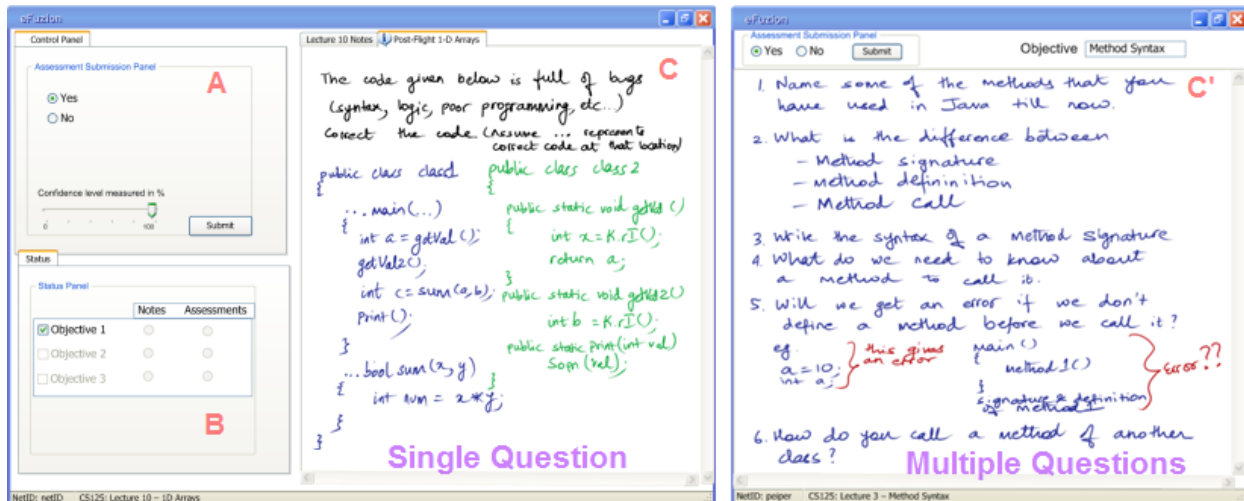


Figure 6.17: Updated design (on the right) of post-flight assessment user interface (on the left)

decided that by the time a post-flight was performed, students should have had a good enough understanding of the topic to know whether they could solve the problem. It is analogous to a student who goes through all the sheets of an examination to see how they should proceed in solving the questions. Which questions will take longer? Are there questions I simply cannot answer? Their ability to make such judgments was no different in our class. We therefore reduced the post-flight interface to include a simple “yes” or “no” response and removed the confidence measurement in order to simplify the decision making process. The modifications made to our first design of the post-flight assessment are illustrated in figure 6.17. Overall we simplified the interface, maximized screen real-estate, while retaining the name of the current objective (the subject of the assessment).

The post-flight provided students with an opportunity to receive instant feedback on whether they had enough understanding of a topic to answer a question with similar difficulty on the exam. As a result, we changed the proposed interface to more explicitly state that the student could or could not answer the question. We also modified the user interface to provide ample space for students to articulate how they would go about answering the post-flight question. There were two reasons that we did not require students to solve the post-flight questions in class; firstly there was simply not enough time for students to solve a question that was designed for an exam. The second reason was that we wanted to encourage students to complete the exam-like questions outside of the classroom as a means of reinforcing what they learned in class. Regardless of a student’s answer as to whether they could or could not answer the post-flight question, students were encouraged to take responsibility for the feedback received within the lecture, outside of the classroom. Figure 6.18 illustrates the first interface implemented for the spring 2007 pilot study. In this example, the student knows the current objective, they know they are in “assessment” mode, they have

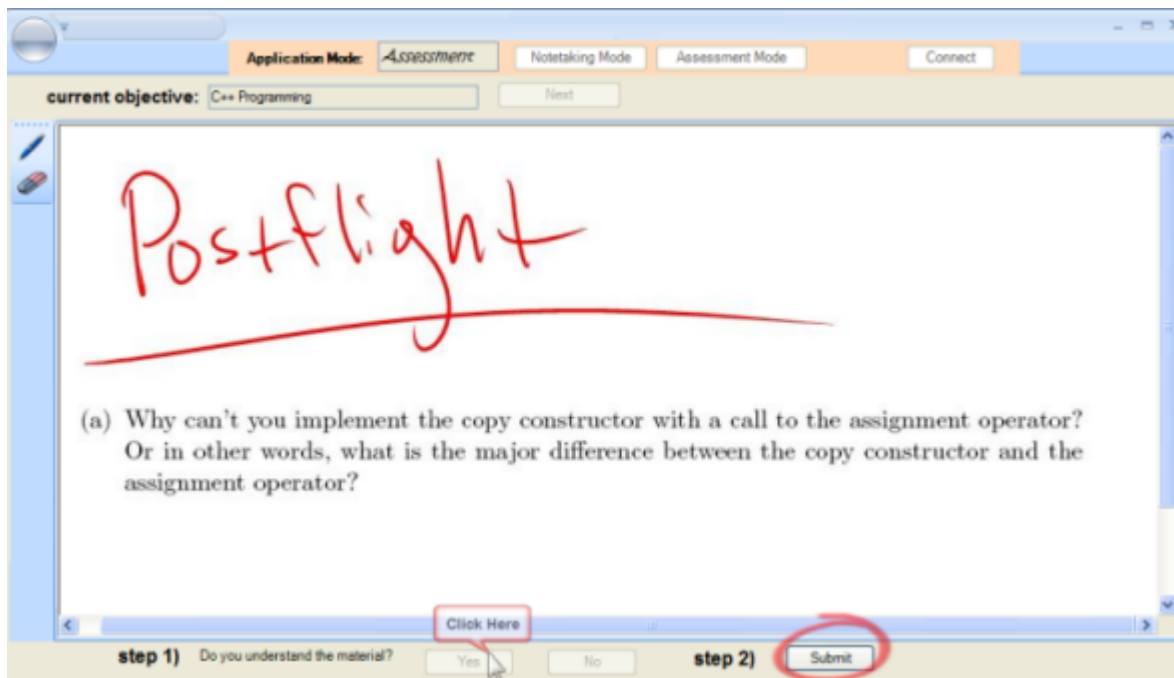


Figure 6.18: Post-flight interface used in the spring 2007 pilot study

already clicked “yes” for step one, and have not begun to ink a response before submitting their answer.

While the system had the ability to measure a student’s interaction with the Tablet PC, the focus of this study was to provide the instructor with an aggregate of student responses in real-time. Page information such as ink usage was recorded, however, the important aspect of the pre-flight was the result sent by the student to the instructor; “yes I believe I understand the material being presented in the current objective” or “no, I need to pay attention to the lecturer.” We considered the idea of setting a time limit (some static value determined by the instructor) for the students to submit their responses. Should we place added time pressure on students when all we wanted to know was whether or not the student understood the material? Under such a time constraint, what should an answer default to if a student did not submit their response within the allotted time? Because we allowed for students to work somewhat independently of the instructor it was possible that students would be taking a pre-flight asynchronously. The motivation for placing a time limit on the pre-flight was to allow the instructor to get a rough idea, as quickly as possible, of the percentage of the class that “needed to pay attention” (that is, how many students planned on attending the ten minute lecture talk on objective A at time X?) One way of dealing with this problem was supporting the inclusion of a time limit when an instructor initiated a timer through the feedback control panel. When the instructor wanted to “force” the students to take a pre-flight (only those students who did not already perform the pre-flight) or a post-flight (everybody in lock step) she did so by the push of a button as illustrated in

figure 6.11.

6.3 Explicit Polling of Students

This system required explicit interaction by students before and after performing all assessments. For students to submit their pre-flight and post-flight responses, they had to select either “yes” or “no” followed by a click of the submit button. If a student wanted to attempt another assessment or move on to a subsequent objective, they did so by pushing a button. Although this type of system interaction required explicit user submission and navigation, it was less obtrusive than an on-the-fly pop-up dialogue which had the potential to interrupt a student’s current activity. As we report in chapter 7, implicitly capturing user interactions for the purposes of providing real-time feedback without requiring explicit interactions by a student is an example of un-obtrusive [193] capture of system interaction.

In the fall 2004 section of CS400, we attempted to integrate explicit feedback as a means of providing additional information for the logs recording the writing attributes of the students. The students were instructed to click one of the three smiley faces to document their current “mood” in the log files. We found that placing the responsibility of the students to provide explicit feedback was not very effective. In this study we took the opposite approach in that we explicitly sent feedback polls to students currently in note-taking mode. (We did not want to interrupt the students in assessment-mode.) The three questions included in the poll, were in part, related to Marzano’s Taxonomy of Objectives [124] and included questions that quickly queried a student’s level of interest, understanding, and pace (figure 6.19).

The poll results (figure 6.20) for the explicit feedback dialogue were displayed in a simple text-based form where the values for each answer (a rating of one to five) were accumulated as the students submitted their feedback. In the event that a student chose not to answer the poll, clicking submit sent default zero values for all three questions (“I am currently interested”, “I am currently understanding”, “The instructor is going too slow or too fast”). While the utility of the real-time explicit poll was to provide an instructor with data on the students’ level of interest, understanding, and pace, another motivating factor for recording student responses was that we could correlate the student responses to inking behaviors. Does the student write more quickly when they are bored, with more pressure when they are frustrated, and perhaps not at all when uninterested? A more suitable visualization for reporting the results of the explicit feedback poll is illustrated in figure 6.21. The focus of this pilot study was the real-time monitoring of student interactions. The explicit polling was secondary to the implicit information being recorded in log files such as page attributes as discussed in section 6.2.5.

Feedback

Please tell us your thoughts about the current topic in the lecture
(answer as many or as few as you like)

I am currently interested (not at all=1, very=5):

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

I am currently understanding (nothing=1, everything=5):

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

The instructor is going (too slow=1, too fast=5):

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Submit

Figure 6.19: Explicit feedback poll sent to students in note-taking mode

	Interest	Follow	Speed
0	0	0	0
1	0	0	0
2	0	0	0
3	1	0	1
4	0	1	0
5	0	0	0

Figure 6.20: A) Template for explicit feedback poll result B) An example of poll results for a single student

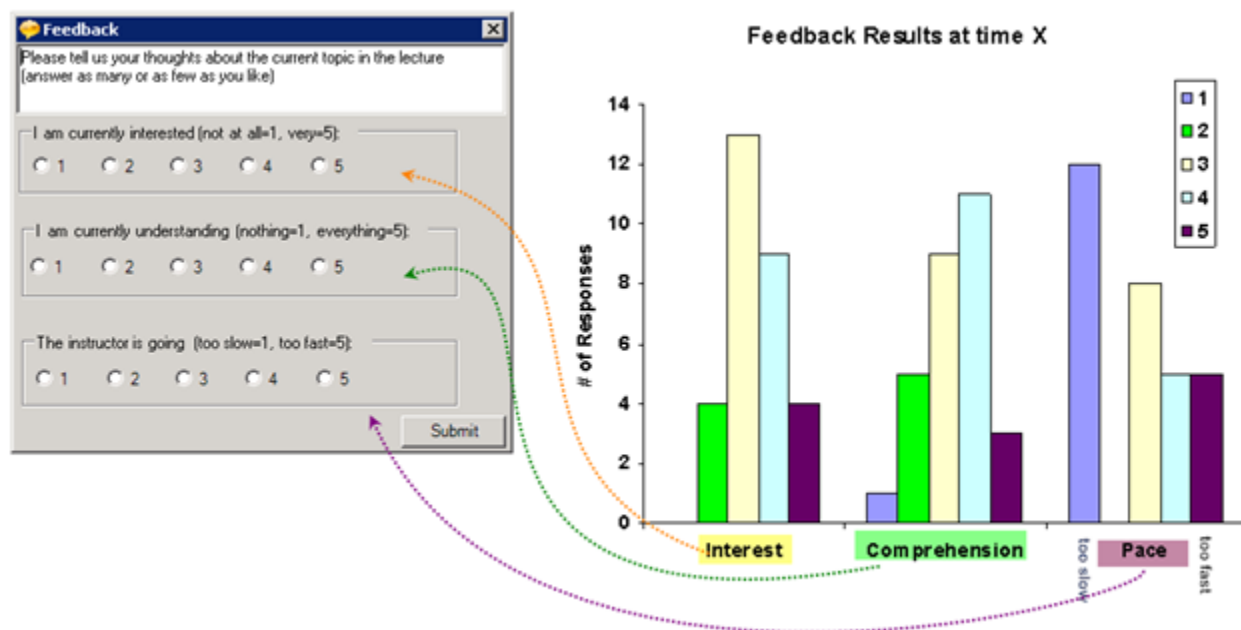


Figure 6.21: At key points during the lecture, polls were automatically sent requesting students to articulate their “state of mind.” The idea here was that an instructor could look at the results of this query in real-time and adjust their teaching accordingly. The graph suggests that students were mildly interested in the current topic, had a good understanding, and believe the pace was too slow.

In figure 6.22 we show a hypothetical example of how we could drill down into our classroom data to gain an understanding of student performance of objective three, exercise one. Imagine that the instructor is going to teach objective three, three different ways in periods of five minutes. So, the instructor has planned to spend 15 total minutes on this particular objective. The idea here is that we can see how many students understood the objective before, during, and after the three “mini” teaching sessions.

While this information could be automatically generated after the class session, during the lecture (in real-time) an instructor could easily gauge the level of engagement (i.e. how many students have submitted assessment one of objective three). That is to say, after an instructor attempts to teach a particular topic, what are the current ratings (how many people are “tuned” in?) This allows an instructor to make adjustments in real-time, reflect on future lectures, and provide students with feedback on their own performance.

It is interesting to note that one student submitted the assessment outside the time frame dedicated to objective three (minute 21 on the right-hand side of figure 6.22) while the instructor had already moved on to objective four. This would be reflected (verified) by looking at the current activity of students. Specifically, all but one student are either listening to the instructor and “taking notes” or performing assessments related to objective four. This type of information illustrates how easy it is to process data representing the outliers (the students not in the “norm”).

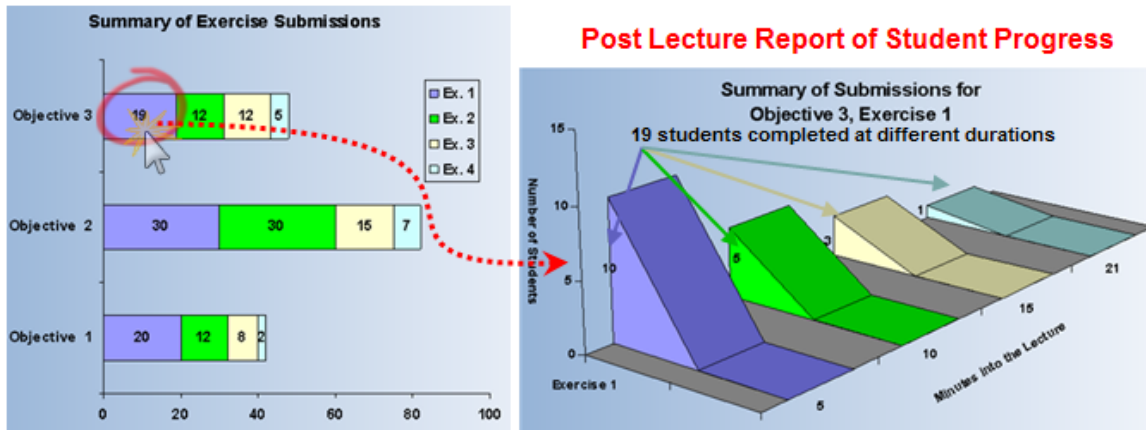


Figure 6.22: Example of “drilling down” into the graph. By clicking on the first bar of objective three (the inscribed 19), we could get a visualization at the point in the lecture where the 19 students who completed objective three exercise one did so.

6.3.1 Teacher display

The teacher’s display was our first design attempt of a real-time dashboard of student interactions. The system retrieved student information by a polling mechanism, in which the client applications sent messages to the server whenever their status changed. The novelty of this system was the real-time display of student interactions with the Tablet PC devices. Example questions we wanted to report included but were not limited to the following: How many students were attempting assessments? Which objective assessments were they working on? How long have they spent trying to solve an assessment? Were there students in note-taking mode? Have these students completed any assessments, answered “no” to the current objective pre-flight, gone outside the scope of the application (for some threshold of time)? Figure 6.23 is an example dashboard for a lecture that included three objectives on binary search trees: introduction of the binary search trees (BST), binary search tree insert algorithm, and the binary search remove algorithm.

6.4 Student Evaluations

At the conclusion of the spring 2007 pilot study we surveyed 17 students enrolled in the Tablet PC section of CS125. We administered the same questionnaire that was presented in chapter 5 section 5.4 regarding the use of instructor-created whiteboard notes and students’ perception of how technology effected the learning environment. Recall that the surveys completed in the large lectures of CS125 and CS225 were taken from the perspective of students *observing* an instructor using a Tablet PC. Since all of the students in the spring 2007 pilot study used Tablet PCs during class we wanted to compare the survey results between the two

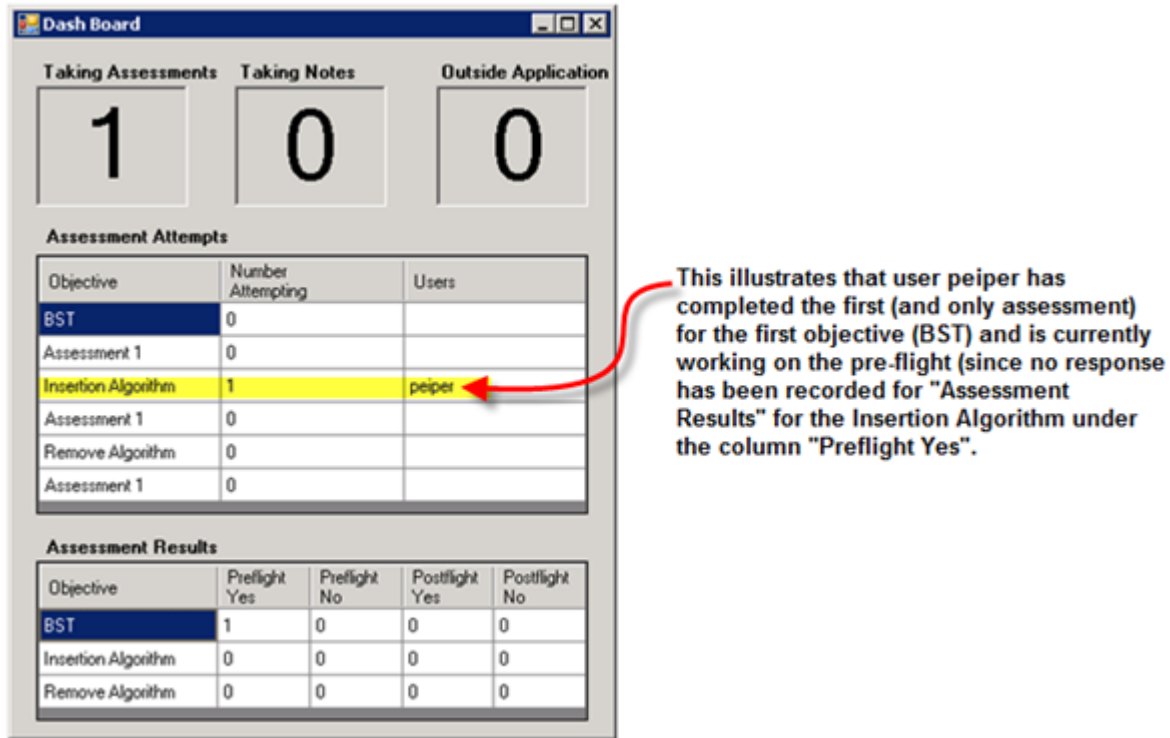


Figure 6.23: An example of our first teacher's dashboard design

sections. In this section we summarize the overall results and highlight some of measurements of interest². A table of survey results for questions requiring one of the five measurements (Strongly Agree, Agree, Neutral, Disagree, and Strongly Disagree) are displayed in figure 6.24.

- Q2 The classroom lecture was more engaging or interesting to me as result of the use of technology in class and the availability of notes afterwards.
- Q3 Because captured lecture notes are available after class, I was able to better pay attention to the lecture.
- Q4 The lecturer used the e-Fuzion technology effectively in class.
- Q5 This type of course is well-suited to e-Fuzion technology.
- Q11 All things being otherwise equal, I would prefer to take a class that uses e-Fuzion technology over the same class that does not.
- Q12 e-Fuzion technology will encourage students to skip lectures.
- Q13 e-Fuzion made me less worried about missing class.
- Q14 I trust that captured lecture notes will be available after every class.
- Q15 Audio augmentation of the Web-based lecture notes would have increased their value to me.
- Q16 Video augmentation of the Web-based lecture notes would have increased their value to me.
- Q17 Printing captured lecture slides after class for later review was valuable to me.
- Q18 The value of captured lecture notes would be greatly enhanced if replay of the instructor's ink was possible.
- Q23 The availability of prepared lectures (printed out) before class would make it unnecessary to access the captured notes after class.
- Q24 If you accessed the captured notes, were you more likely to print them out instead of view them online?

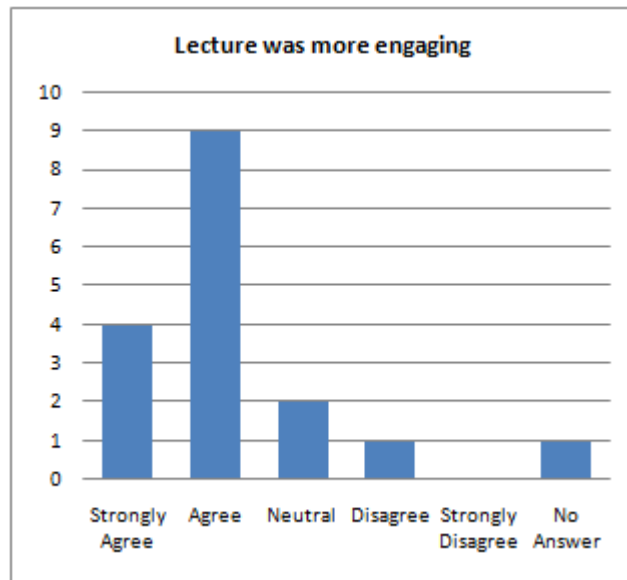
²Questions 19, 20, and 21 were omitted from the survey

	1	2	3	4	5	NR	
Q2	4	9	2	1	0	1	
Q3	2	2	10	2	1	0	
Q4	3	12	1	0	1	0	
Q5	3	8	3	2	1	0	
Q11	3	9	2	0	1	1	
Q12	1	2	8	4	0	2	
Q13	0	4	5	4	3	1	
Q14	2	9	3	1	0	2	
Q15	0	4	8	2	1	2	
Q16	0	6	5	3	1	2	
Q17	2	5	5	3	0	2	
Q18	5	5	3	3	0	1	
Q23	2	4	7	3	0	1	
Q24	2	4	10	0	0	1	

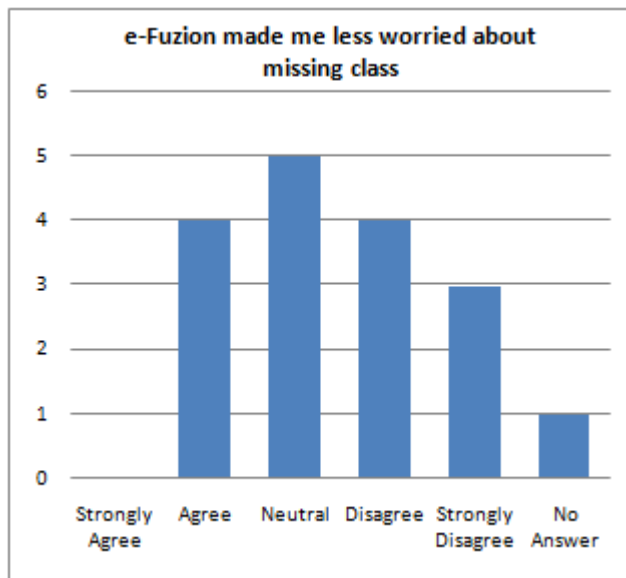
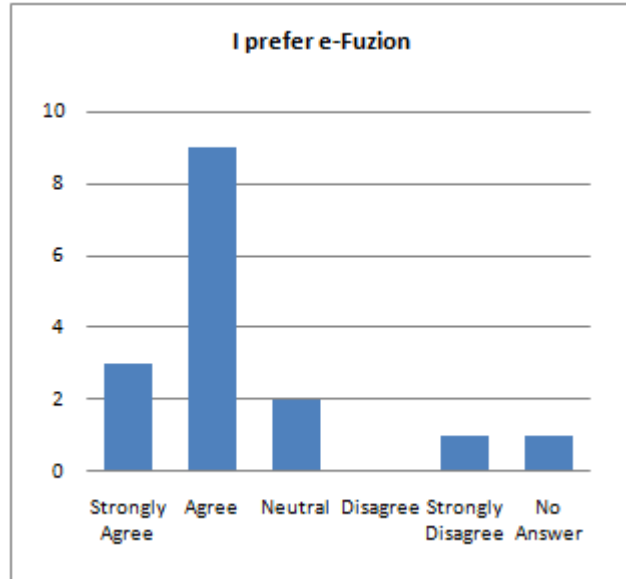
1: Strongly Agree
2: Agree
3: Neutral
4: Disagree
5: Strongly Disagree
NR: No Response

Figure 6.24: Spring 2007 pilot study survey results for questions 2–5, 11–18, and 23–24

Similar to the survey results presented in chapter 5, students reacted most positively to the instructor’s use of the e-Fuzion technology³. In contrast to the results from the large lectures, however, students in this study reacted more positively in their evaluation of lecture engagement (question 2), preference for using e-Fuzion (question 11), and dependence on being *in-class* (question 13).



³Although the application integrated in the study was NuPaper®, the directions on the general survey referred to the technology used in-class as “e-Fuzion.”



6.5 System Implementation

6.5.1 System architecture

The user interface used in the study was written by Boris Capitanu of NuPaper®. The network protocol integrated into the system was called Sharecast® and was designed and written by David Warden. Both Boris and David are recent graduates of the Department of Computer Science. The Sharecast®(section 6.5.5) was an implementation of the Pragmatic General Multicast (PGM) protocol (section 6.5.4) which is a reliable multicast transport protocol [60]. PGM runs under the User Datagram Protocol (UDP) which is also known

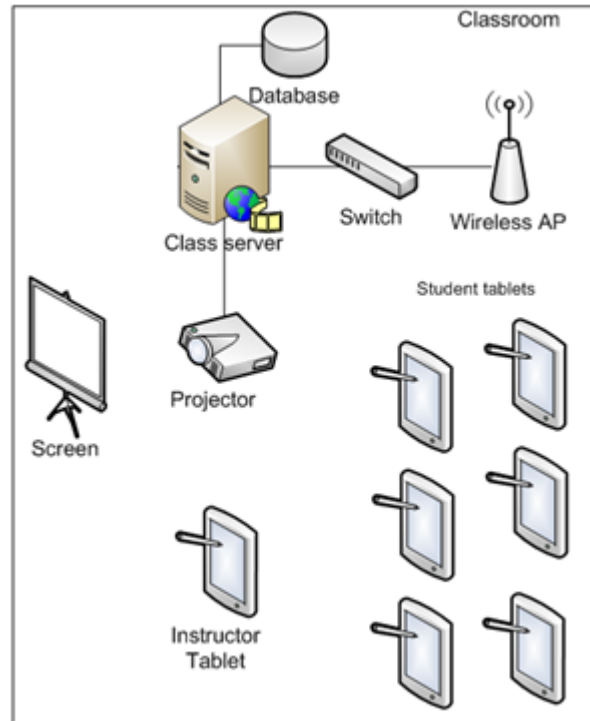


Figure 6.25: Illustration of the system architecture for the spring 2007 pilot study

as a “best effort datagram service” [145]. The novelty of Warden’s system was that it implemented a reliable **wireless** multicast (see section 6.5.2). This amounts to a promise that all wireless network messages will eventually be delivered to server and client machines. For a good explanation of unicast and multicast see Bristow [23]. The system architecture is illustrated in figure 6.25. The technical components have been abstracted out and are discussed below.

6.5.2 Wireless networking challenges

Empirical testing has shown that when used exclusively with traditional unicast TCP/IP networking protocols, common 802.11a/b/g wireless networks are insufficient for simultaneous transmission of classroom data to more than twenty-five machines. The reasons for this are several-fold. Wireless networks suffer from greater packet loss than wired networks due to greater electromagnetic interference. While wired networks use packet-switching hardware to direct packets only to intended recipients, wireless channel bandwidth must be shared among all machines. Finally, wireless throughput is typically less than that found on wired links. Current 802.11a networks have a maximum speed of 54Mbps whereas wired link generally have 100-1000Mbps. (Even with the forthcoming 802.11n standard, typical wireless bandwidth will remain below that of wired networks.) At the same time, users expect more bandwidth-intensive experiences, including more

Multicast Transimission
Uses bandwidth more
efficiently but makes reliable
communication more difficult.

How is this problem dealt
with?

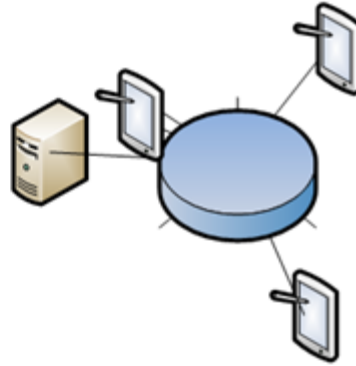


Figure 6.26: Multicast transmission illustration

high-resolution graphical and multi-media content.

6.5.3 Multicast solutions to networking demands

Multicast data transmission allows data to be transmitted once and received by multiple clients, therefore avoiding duplicate communication. Unfortunately, it is difficult to verify that the data has been properly received by all clients. For some applications, such as video transmission, lost packets are simply tolerated by the application. However, for inking and other applications in the classroom environment, such an approach is inadequate; the communication must be made 100% reliable (figure 6.26).

Multicast communications may be made reliable using a **negative acknowledgment** scheme. Using negative acknowledgment, clients transmit a notification when they do not receive a packet they are expecting, which may be detected when another packet is received outside of an assigned sequence. To insure clients are up-to-date even when no new information is available, **heartbeats**, or packets without additional data, may be used by the protocol or application to alert clients of the current sequence number.

6.5.4 The Pragmatic General Multicast

A proposed standard, the Pragmatic General Multicast, or PGM, is defined in RFC3208 [60]; see also <http://www.faqs.org/rfcs/rfc3208.html>. PGM uses a negative acknowledgment scheme with features to prevent nack implosion, wherein a lost packet causes all clients to send a negative acknowledgment at the same time, overwhelming the transmitting servers. When using PGM, clients listen for negative acknowledgments from other clients during an interval of time having a pseudo-random duration. If another client requests retransmission, other clients delay their request until such a retransmission should have been received. PGM also allows for hierarchical transmission wherein intermediate devices, such as routers, can combine retransmission requests for a number of clients. Implementations of PGM exist for recent releases

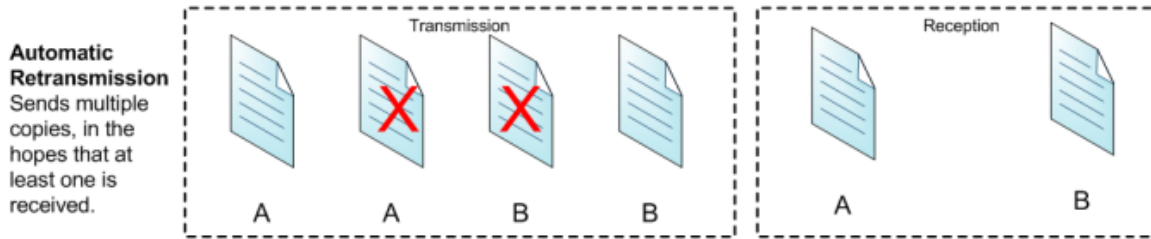


Figure 6.27: Illustration of automatic retransmission of packets sent over a wireless network

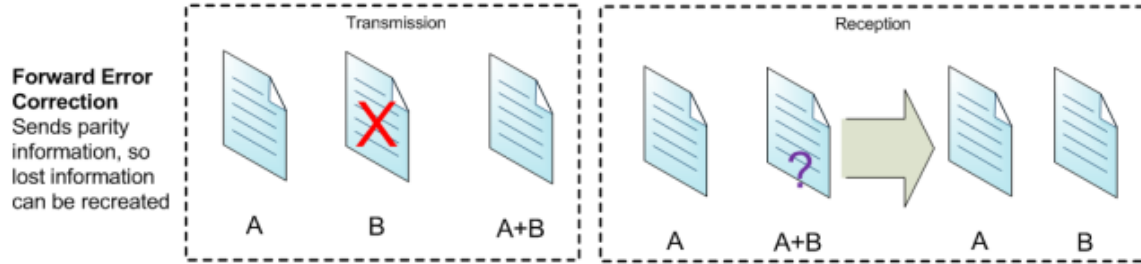


Figure 6.28: Illustration of forward error correction (FEC) mechanism for packet delivery

of the Microsoft Windows platform, along with implementations embedded in certain Cisco routers.

PGM contains several features to improve the performance of multicast transmission, which are particularly suited for wireless environments. Automatic retransmission (figure 6.27) allows packets to be transmitted multiple times on the presumption that packets will be lost. Forward error correction (figure 6.28) uses a parity scheme or error correcting code to allow a group of packets to be reconstructed based on reception of a smaller subset. However, PGM is not without certain limitations. PGM retransmissions may occur only within a fixed-size window. Clients joining late, or suffering network problems, may not be able to obtain required data from PGM. Certain network topologies—such as the Internet—may discard multicast packets, and as a result cannot be used with PGM. In any case, obtaining high performance from PGM sessions on wireless networks requires careful attention to protocol parameters.

6.5.5 The Sharecast® networking protocol

The NuPaper Classroom® application used in this study made use of a higher-level multicast-capable networking protocol, Sharecast, for several purposes. Sharecast is a session-oriented reliable communications protocol optimized for wireless environments. Sharecast uses PGM to transmit shared data whenever possible, but automatically reverts to unicast transmission when necessary. The Sharecast protocol is implemented in C# .NET on Microsoft Windows.

Using Sharecast, clients use TCP to transmit XML-based messages to a server where they are sequenced

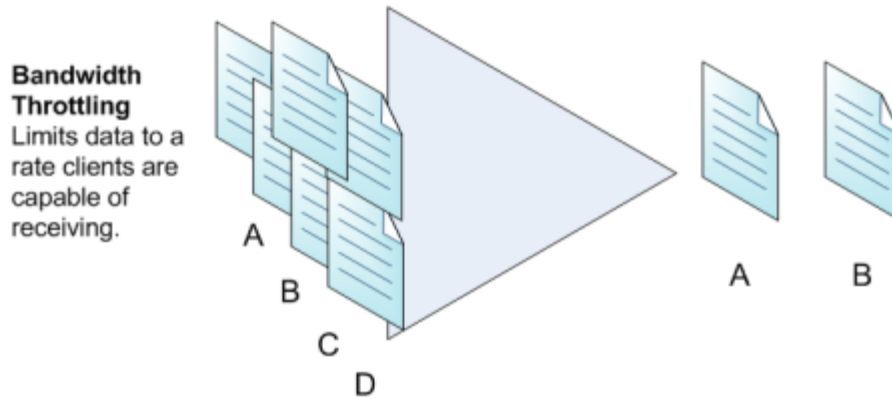


Figure 6.29: Illustration of bandwidth throttling for limiting data rates to be compatible with client connections

and forwarded as needed. In contrast to peer-to-peer based approaches, a client-server architecture makes it easy to derive a consistent view of session information.

Sharecast clients are always in one of two operating modes, unicast mode or multicast mode. Clients connect in unicast mode, but switch to multicast mode whenever possible, as it is more bandwidth efficient. PGM data is buffered, and when unicast data is received with sequence numbers up to the buffered segment, the client may transition to multicast mode. If the PGM session is lost, the client transitions back to unicast mode, buffering new unicast data and receiving any missing data through a request to the server. Non-sequenced data intended for a single client is also unicast, and the server transmits in unicast mode until multiple clients connect. Any number of unicast-multicast transitions may occur during a Sharecast session.

Sharecast uses Pragmatic General Multicast with automatic retransmission and forward error correction enabled to compensate for packet loss in wireless environments. Sharecast performs data compression, optimizes packet size, and throttles transmission rates (figure 6.29) for optimal performance with PGM on 802.11 wireless networks.

6.5.6 Sharecast® in the spring 2007 pilot study

Sharecast was used for the transmission of assessments and status information. Having had a number of years of experience dealing with the problems of network latency and network contention, we sought to reduce the problems usually accompanied with wireless transmissions in larger-sized classrooms. Recall that the feedback console (illustrated earlier in figure 6.11 and called the feedback control panel) was first used to create a class session which resulted in figure 6.31), load assessments consisting a PDF document and associated XML metadata (see figure 6.30), which are transmitted to students at the start of class.

As the students in the class completed assessments their progress was transmitted to the console dash-

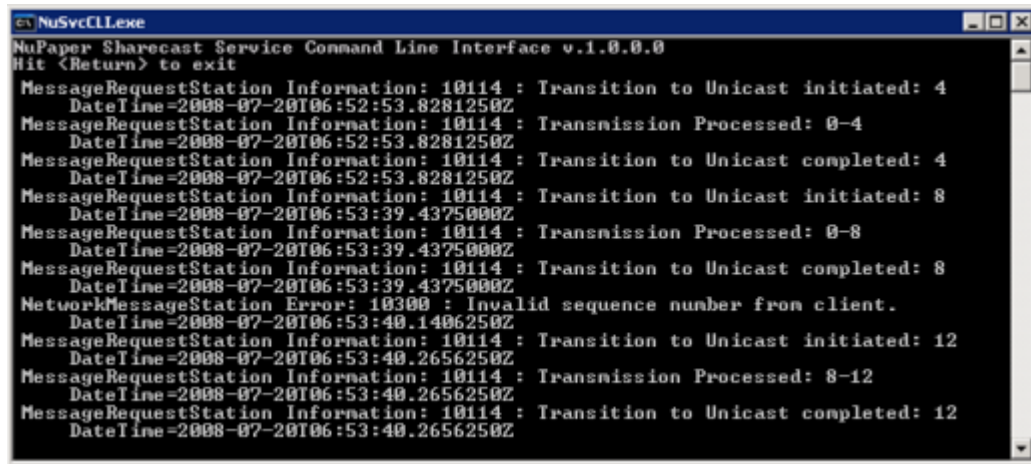


```

B
<?xml version="1.0"?>
<objectives>
<objective name="BST" file="bst.pdf"/>
<objective name="Insertion Algorithm" file="bstInsert.pdf"/>
<objective name="Remove Algorithm" file="bstRemove.pdf"/>
</objectives>

```

Figure 6.30: The XML schema contains three objectives and expects three PDF files



```

NuPaper Sharecast Service Command Line Interface v.1.0.0.0
Hit <Return> to exit
MessageRequestStation Information: 10114 : Transition to Unicast initiated: 4
DateLine=2008-07-20T06:52:53.8281250Z
MessageRequestStation Information: 10114 : Transmission Processed: 0-4
DateLine=2008-07-20T06:52:53.8281250Z
MessageRequestStation Information: 10114 : Transition to Unicast completed: 4
DateLine=2008-07-20T06:52:53.8281250Z
MessageRequestStation Information: 10114 : Transition to Unicast initiated: 8
DateLine=2008-07-20T06:53:39.4375000Z
MessageRequestStation Information: 10114 : Transmission Processed: 0-8
DateLine=2008-07-20T06:53:39.4375000Z
MessageRequestStation Information: 10114 : Transition to Unicast completed: 8
DateLine=2008-07-20T06:53:39.4375000Z
NetworkMessageStation Error: 10300 : Invalid sequence number from client.
DateLine=2008-07-20T06:53:40.1406250Z
MessageRequestStation Information: 10114 : Transition to Unicast initiated: 12
DateLine=2008-07-20T06:53:40.2656250Z
MessageRequestStation Information: 10114 : Transmission Processed: 8-12
DateLine=2008-07-20T06:53:40.2656250Z
MessageRequestStation Information: 10114 : Transition to Unicast completed: 12
DateLine=2008-07-20T06:53:40.2656250Z

```

Figure 6.31: Upon creating a class session using the feedback control panel illustrated in figure 6.8, Sharecast service provided an implementation of a reliable wireless multicast. For the two semesters that we ran the pilot study, we never once had a problem such as network failure or latency. Quite to the contrary, we were able to send lecture documents and receive almost instantaneous feedback from students performing assessments. While glancing at the teacher's dashboard, we could easily detect when students went outside the application (which we considered as sign of being off task) immediately.

board (figures 6.8 and 6.23). The state of each student's machine (specifically whether they are in note-taking or assessment mode, or outside of the application; along with their assessment results) was transmitted to the console in real time. Poll data (measuring students understanding, interest, and satisfaction with the speed of the lecture, figure 6.19) could be sent on demand or based on expiration of a timer, with results updated in real-time (figure 6.20). Finally, the instructor controlled the operation of the classroom in several ways, by beginning or concluding the various assessments related to a particular objective (such as pre-flight and post-flight assessments), using the feedback control panel.

6.6 Conclusion

In this chapter we presented the design and implementation of a system where a classroom structure in the form of lecture objectives and assessments was used to provide feedback to instructor and students.

One of the advantages of the system was that it provided the instructor with the information necessary to facilitate collaboration amongst students [22]. By following the progress of the students as they navigated through the set of lecture objectives, it was possible for the instructor to focus her attention on the students who required additional attention. Students who would normally be off task surfing the web or working on other assignments were easily identified and assigned teaching roles. This type of intervention would not have been possible without the integration of the teacher's dashboard as described in this chapter. Although we did not perform a controlled study, the student participants in the spring 2007 pilot study outscored the students registered for the "regularly scheduled" large lecture of CS125. Student surveys completed at the end of the semester confirmed our observation that students enjoyed learning in this type of classroom environment. Student scores for the second midterm also confirmed our belief that teaching in a constructivist student-centered environment with technology has the potential to improve learning.

Chapter 7

Fall 2007 and Spring 2008 Dashboard Study

The minimum we should hope for with any display technology is that it should do no harm.

Edward Tufte

7.1 Introduction

Networked tablet PCs have great potential in classroom settings, including use in small group in-class problem-solving activities. It is possible to obtain substantial amounts of data about student activity during a lesson: what they referred to, notes taken and erased, bursts and lulls of activity. The raw data is necessarily very low level: time-stamped pen strokes, deletions, navigation to and from pages, to name but a few. This data has the potential to be used in a variety of ways, one of which is enabling a teacher to monitor learning activities as they happen in real-time in the classroom. In investigating how to support this use we uncovered a particularly interesting visualization design space.

The challenge is to provide the teacher with minimal at-a-glance overview information. This information must be capable of being integrated with conventional visual and auditory clues about student activity and progress that the teacher uses in order to make quick decisions about what to do next: which individuals or groups need to be addressed, how have students progressed while the instructor interacted with a group of students. The biggest constraint is that teachers want to spend the bulk of the time interacting with the students and not the system.

In this chapter we present a two semester study (fall 2007 and spring 2008) that explored the use of Tablet PCs in small group in-class learning activities. We describe our first iterations of design and evaluation of our system in an actual classroom setting, and the lessons learned. This kind of use builds heavily on a well established body of activity in many classrooms, both in K-12 education and at the University level in smaller discussion or laboratory section classes.

Typically these class sessions included a series of problem tasks that might be set up on the blackboard

or other public display, or selected from a textbook. However, very often a worksheet is prepared in advance by the instructor and given to the student. How might the well-understood worksheet concept be adapted for use on a tablet, and how might the technological affordances of networked tablets add value to this technique?

In investigating these questions we uncovered a fascinating design space of constraints and opportunities that require particular kinds of visualization approaches that are typically not considered in more conventional scientific and information visualization settings.

In section 7.2, we review related work that informed our designs and also show how our work fits into the larger picture of educational computing and visualization research. Section 7.3 outlines the design space. In section 7.4 describes our first iterations of design and evaluations of use in actual classroom settings, and the lessons learned. Section 7.5 outlines the new design approaches currently being developed that have been informed by that analysis. In section 7.6, we discuss four dimensions of dashboard content, and their implications for dashboard design. In section 7.7, we describe the actual dashboard design we used in this study. In section 7.8, we present the actual results of our study in the form of student evaluations.

7.2 Related Work

Visualization has a long history in the classroom. Although blackboard and chalk illustrations have given way to modern multimedia presentations, the idea of using visual aids to enhance student knowledge acquisition is not new. What is new is the idea of applying information technology and visualization to enhance an educator’s comprehension of how a class is learning and performing. The system described in this chapter is essentially unique in addressing this task. Simoff and Maher [167] describe how data mining can be employed in aiding student assessment in collaborative learning environments. While our work has a similar goal, our approach was to employ information visualization to aid the instructor in student assessment. Wortman and Rheingans [202] used visualization to assess student progress across courses through a curriculum as opposed to the in-class assessment our system provides. Among commercial products, the DyKnow™ Monitor® and Vision® (see section 2.8.3) and NetSupport School (<http://www.netsupportschool.com/overview.asp>) systems both allow an instructor to see the computer screens of students and monitor their activity. However, neither of these systems tracks detailed information regarding student interaction (e.g. amount of electronic ink usage and amount of erasing) as does the system described in this chapter. Moreover, neither system presents a statistical visualization of recent student activity as we propose in this chapter.

Dr. Kimberle Koile has advised a number of recently-published theses (<http://projects.csail.mit.edu>).

`edu/clp/publications/#theses`) that investigate these topics within the context of the CLP (see section 2.8.1) application [30, 146, 178, 203]. Smith’s work on aggregation of student answers in a classroom setting [171] is of particular interest and helps motivate the work presented in this chapter.

Our approach to designing a student activity visualization system has been greatly informed by Shneiderman [164] who suggests the “overview first, zoom and filter” framework we employ. The visual appearances of the student activity thumbnails in our system recall the approach of Chi et al. in their work on visualization spreadsheets [31]. Our overall choice of a minimalist approach owes a debt to the work of Tufte [186–189] in general and specifically to his description of sparklines and micro charts.

7.3 The Design Space of Visualizations to Support Tablet PC Use

Using networked tablet PCs it is possible to obtain substantial amounts of data about student activity during a lesson: what they looked at, what they went back to, notes taken and erased, bursts of activity, periods of inactivity, etc. These data can be used in a variety of ways including formative and summative evaluations of Tablet PC usage, for future design iterations of the whiteboard software, and more generally for educational research. This includes observing and comparing students’ learning at various scales of detail from short microteaching episodes of a few minutes to whole lessons and whole courses. Furthermore this data can also be used for supporting trainee teachers by sitting with a mentor and replaying classroom episodes to review their teaching effectiveness, enabling more skilled teachers to assess their own performance and act as reflective practitioners, and to consider why certain interactions were or were not successful for different kinds of settings and students. This allows for experimentation with new techniques and rapid review of initial effects, and enables teachers to monitor learning activities as they happen in real time in the classroom setting.

It is the last case that this chapter focuses on. Note that all settings may use the same raw data requiring only different kinds of processing and visualization to meet the different needs and particular time constraints. In the case of a teacher who has assigned the students to work together in groups on a sequence of tasks, there is a classic set of choices to be made: how is the class as a whole progressing, how are the different groups progressing? Are some people racing too far ahead or lagging too far behind? Are some people and groups performing differently from what the instructor expected of their relative rate of progress? Who needs my attention right now? While I was attending to that person, what was everyone else doing?

Skilled teachers manage to constantly update their answers to these questions in real time, while dispens-

ing advice, help, reassurance and admonishments to students. They use a host of clues including their prior knowledge of the students, their understanding of the tasks, their expectations of likely progress and likely difficulties, overhearing particular conversations and overall background conversation, body language, glancing at the worksheets from afar, etc. Understandably this can be daunting for a novice teacher. It is equally daunting for a visualization developer attempting to replicate and improve upon this feat. Fortunately, this is not necessary. We do not need to replace the teacher’s external clues. We need only to supplement them with additional data in a useful and usable form. That is the core of the design challenge: How to develop a visualization that a teacher can integrate into his or her own existing methods of managing small group learning interactions?

If such visualizations cannot be integrated easily, the system is unlikely to be adopted and experimented with and gradually improved over time. If it serves as a “trust me” application that requires a radical change in teaching practice, many teachers will not adopt it, and those that do will often be disappointed because it will not have gone through the thousands of rounds of iterative improvements of traditional teaching practices.

Given all the other clues that the teacher uses, and given the understandable desire to spend the bulk of the time interacting with the students, not the system, what is needed is an at-a-glance dashboard visualization that can help the teacher make quick decisions about what to do next. It does not have to be perfectly accurate in its predictions (neither are the teacher’s traditional clues), and teachers are very good at recovering from pedagogical errors. It just has to be fast, informative, and act as a useful supplement.

From this analysis, we believe that all visualizations developed need to enable the teacher to answer some (if not all) of the questions outlined above. In subsequent sections we describe various interface features that attempt to do that and discuss our findings about their relative success.

7.4 Preliminary Design and Evaluation

The most exciting prospect of deploying tablets in a classroom is the ability for us to capture and monitor the learning process, regardless of the activity and level of interaction. Our assumption is that students using Tablets PCs in the classroom are engaged in one of the following activities: taking notes, working on an exercise or assessment, or articulating, expressing, and formulating questions, doubts, or uncertainties. If a student has no interaction with the tablet, the system will not have much information on the student (other than the fact that the student has not interacted with the tablet). In this circumstance, the instructor will have to resort to the traditional means of measuring the student’s level of engagement with the lecture

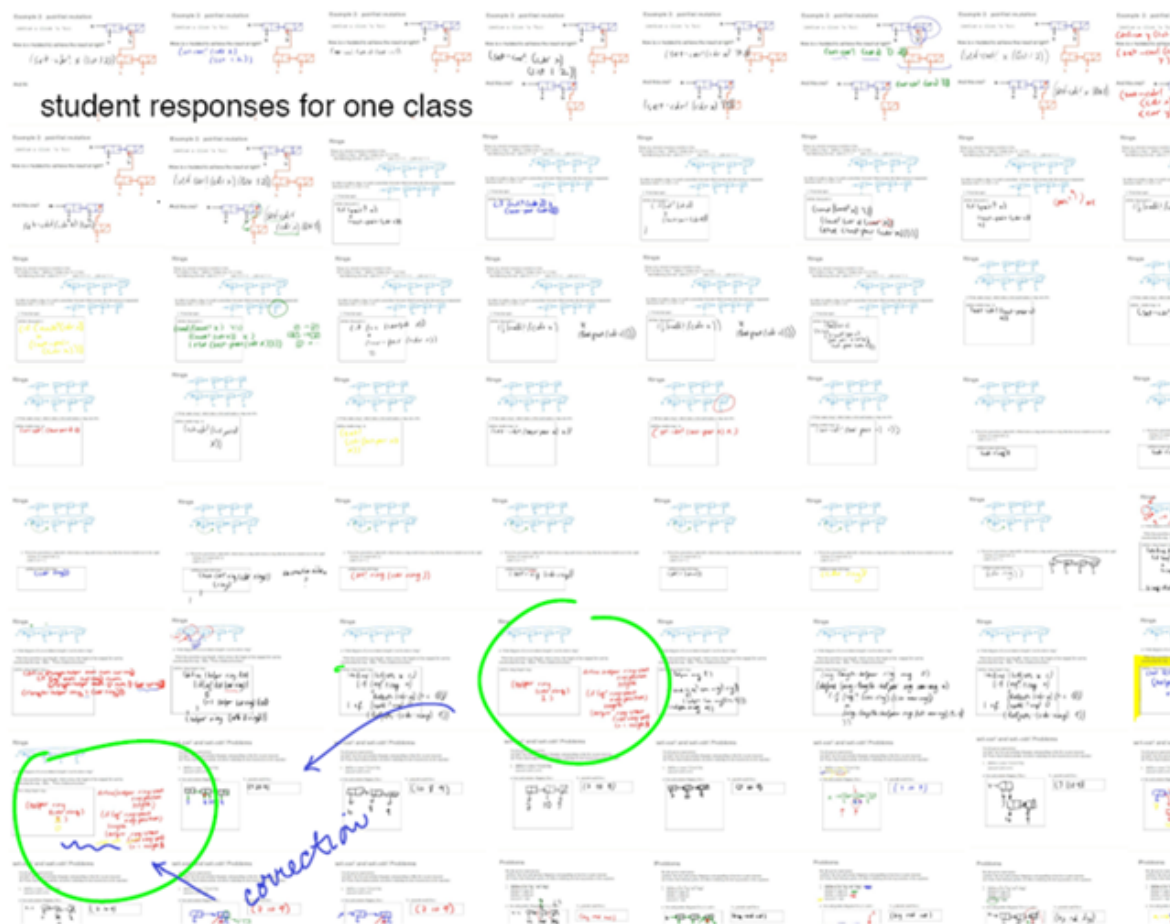


Figure 7.1: An example of a large class of Tablet PC users submitting a response for a single class (source: <http://icampus.mit.edu/projects/clp.shtm>)

content.

The focus of our study was the display of student/course interactions within a class period. Aside from the technical prerequisites such as a wireless infrastructure, a whiteboard inking application and Tablet PC computers there were a number of preliminary design decisions and features that needed to be completed. We needed to build a system that captured, organized, aggregated, and packaged the interactions created within the student whiteboard application. The second feature was a dashboard display. Our notion of a dashboard is a separate display (not the same display the instructor teaches from) that provides information about student activity in the classroom. A motivating factor for exploring this design space is illustrated in figure 7.1. Most tablet-based networked classroom applications provide a means for students to submit work or panels (using DyKnow's terminology) to an instructor. The problem is processing the overwhelming number of submissions in real-time to affect the learning in the classroom.

A challenging task for most, if not all instructors, is the task of trying to gauge student interest and learning during the course of a lecture. Under current approaches, it is very difficult to understand how well the students are grasping the lecture while the lecture is in progress. It is unrealistic to require one individual to observe, assess, and evaluate (monitor) the classroom, while following a strict lesson plan and lecture objectives in an attempt to teach. The ideal situation would be to adjust the lecture midstream to help the students. Currently, teachers monitor and assess their students' progress by a variety of mechanisms, such as observation of student participation, listening to student questions and comments, providing in-class exercises, and of course on a larger scale, graded assessments.

Currently, the main method of classroom assessment is limited interaction with students. Another type of assessment is passive, or unobtrusive. This is the type of assessment that a teacher does by simply looking around the class, or by walking around the room looking at students' individual work, body language such as facial expressions. The best teachers excel at this, but it is an uncertain undertaking, difficult to quantify, hard to learn, and applicable mainly in classrooms where the teacher knows the students very well. We believe the Tablet PC can dramatically enhance a teacher's ability to unobtrusively [193] assess the class when students interact in the lecture with pen and paper replaced by a Tablet PC.

In the Fall 2007 semester we employed the use of VNC to create a display of student windows (figure 7.2). VNC, or Virtual Network Computing, is a graphical desktop sharing system which uses the "remote framebuffer" (RFB) as a simple protocol for remote access to graphical user interfaces. VNC applications transmit the keyboard and mouse events from one computer to another, sending screen shot images of the screen updates from the destination machine to the source over a network. Our VNC-based interface provided no user interaction and was a WYSIWYG display of student workspaces. The inability for the instructor to organize or search "intelligently" any of the student work spaces further motivated the need of a graphical Teacher's Dashboard. The functional constraints and poor performance of the VNC protocol that we experienced in a class of 30 also motivated a system that could scale.

The physical arrangement of the individual VNC connection windows provided little flexibility in terms of user interaction and useful observation. The instructor could not identify the students by the machine name and was not able to manipulate the size and scaling of the individual VNC windows; also no data was being recorded for post-class analysis. This crude model was our way of investigating how we might create a dashboard that had VNC-like functionalities with a decent user interface and searching functionalities (figure 7.3). Unfortunately we experienced processor overhead on the client machines affecting the performance of the whiteboard application.

We attempted experiments with a commercial product, NetSupportTM School as a means of collecting

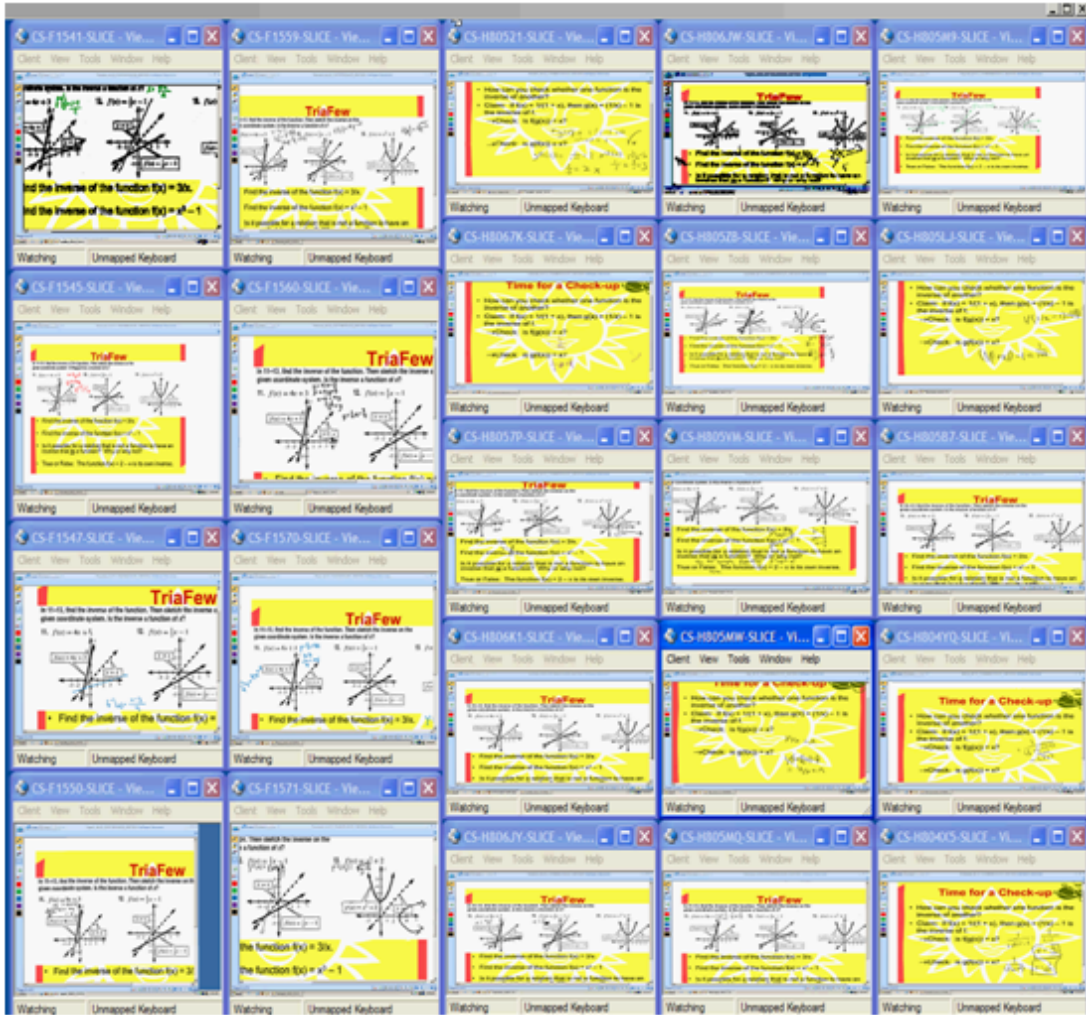


Figure 7.2: An arrangement of scaled VNC connections to 23 client machines in the algebra 1 pilot study

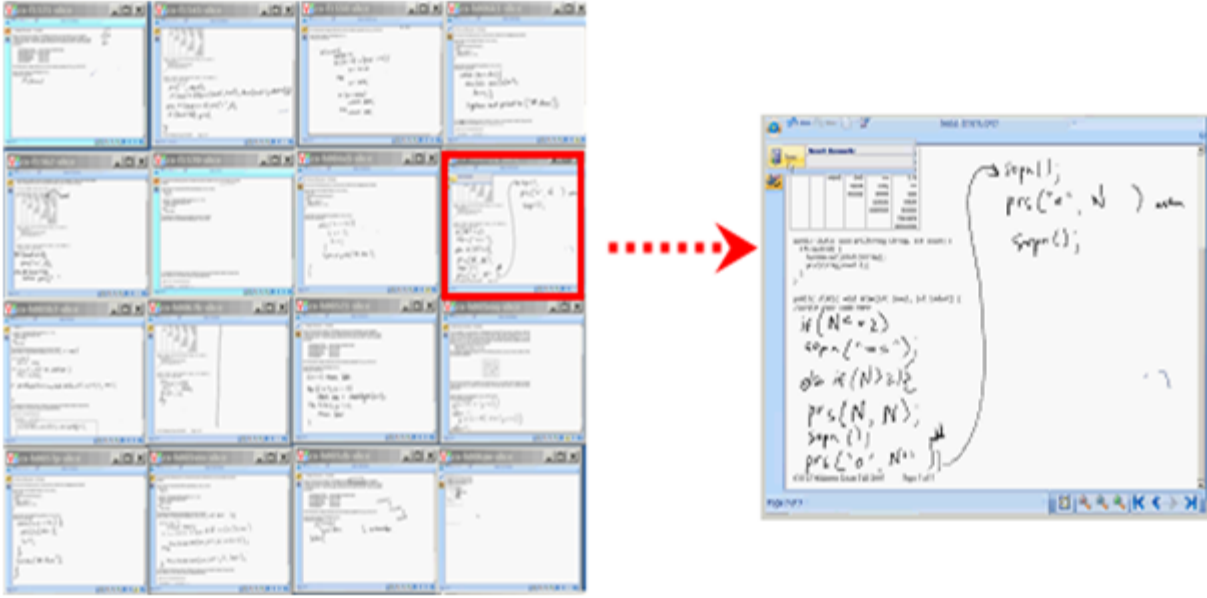


Figure 7.3: An arrangement of VNC 4x4 (16 in total) student windows in CS125 Tablet PC section using NuPaper Classroom® whiteboard application during an exam. One problem that we encountered in this design was processor and network overhead as a result of connecting 24 plus machines via VNC.

similar information to see if we experienced the same machine overload (figure 7.4). Unfortunately, we observed the same problems reported in our earlier design.

7.5 Design Approaches

Having performed a number of pilot studies (chapters 4, 5, and 6) investigating the use of Tablet PCs we refined our study parameters accordingly. The research goals were to design a system to display the **type**, **level**, and **pace** of student activity in the classroom. For example, the pace attribute might provide the instructor with information about whether a student is lagging behind, moving ahead, or following closely with the lesson. How exactly the *type*, *level*, and *pace* are defined within differing classroom structures is one contribution this research addresses. As an example, consider that the instructor might want a real-time comparison of the pace of individual students or an aggregate amongst a subset of students for a given activity. A cursory glance of a good dashboard might point an instructor toward the students requiring special attention.

7.5.1 User interface

In this section we describe the user interface designed to minimize the effects of the technology integration into the learning environment for the fall 2007 and spring 2008 studies. Our research interest was primarily

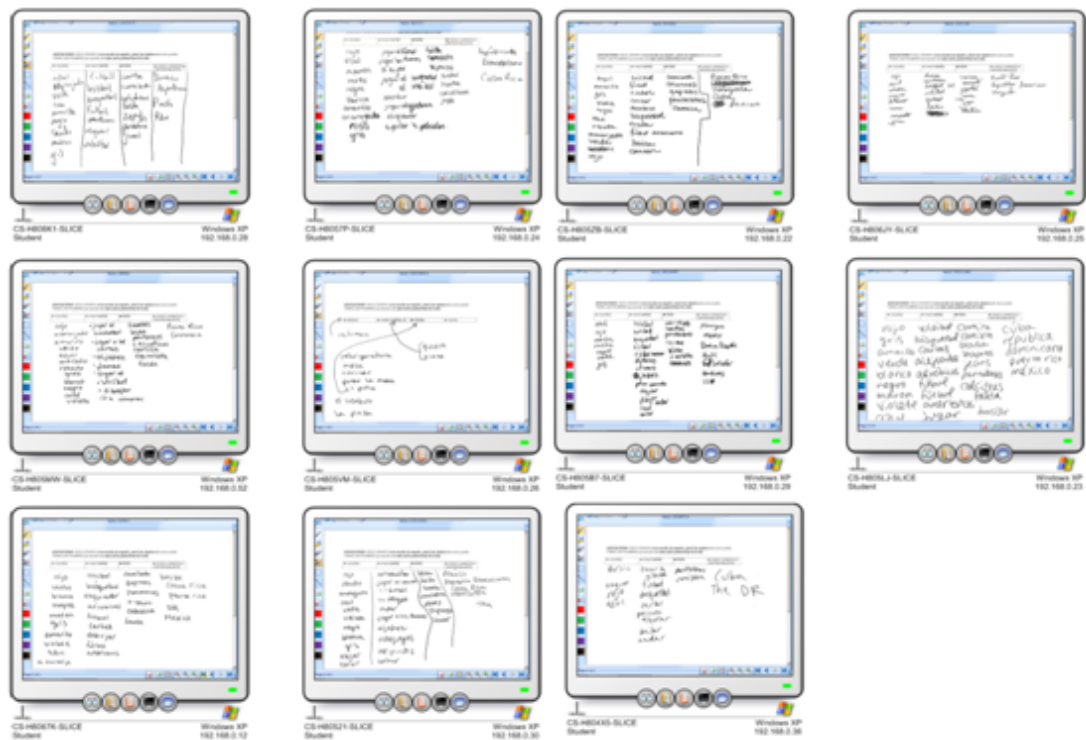
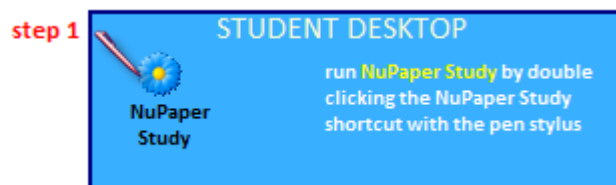


Figure 7.4: A VNC implementation by NetSupport™ where the refresh rate for viewing the client machines could be increased and decreased to one second. This thumbnail VNC interface was a slight improvement from our previous design.

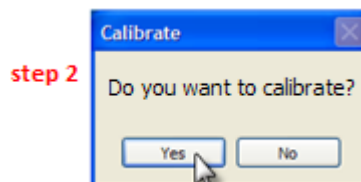
to test our hypothesis that a Teacher's Dashboard could enhance the learning environment. As a result, we wanted to provide as traditional a classroom environment as possible. If the technology was difficult or awkward for the students to use, it would be more difficult to achieve a meaningful outcome. As a result, we wanted to limit the "technology integration" to the instructor rather than to the students. If we could provide a seamless integration whereby students did not have to alter their daily routine as a result of using the technology, we could focus on the instructor's usage of the technology rather than the students' usage. This is an important consideration since the students' interaction with the Tablet PC provided for all of the data we wanted to provide to the Teacher's Dashboard.

One concern was how students could login to the system without requiring the use of a keyboard. Boris Capitanu designed an interface to support a point-and-click manner of selecting user identities (trusting that students would not purposefully login as another student, which the instructor of the course monitored implicitly).

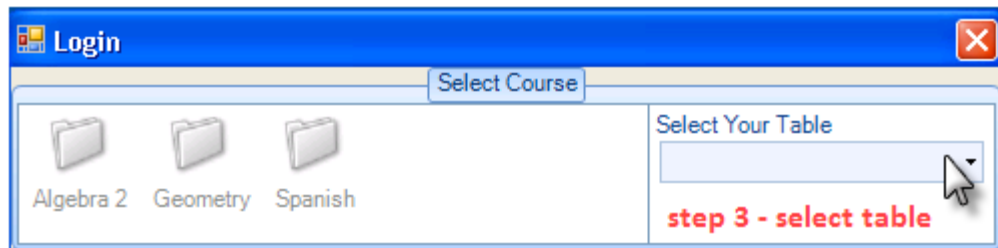
Upon entering the classroom students ran the NuPaper® Study application by double-clicking the icon with the pen stylus



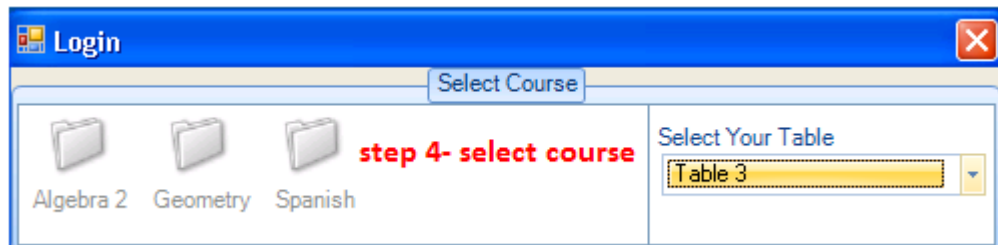
As soon as the application was executed the students were prompted to calibrate their Tablet PC device:



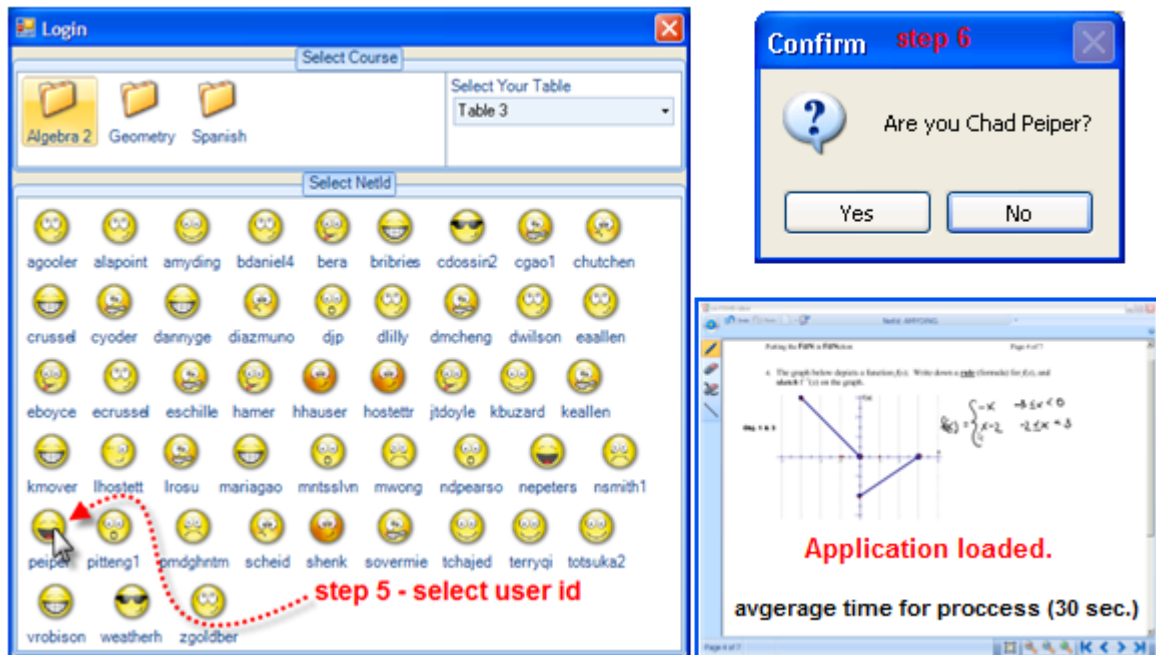
Almost all of the students got into the habit of calibrating their devices since it took no more than five seconds to complete and increased the accuracy of the digitizer. After the calibration was completed students were required to select their table number (the instructor assigned student groups daily according to the level of student understanding of the lecture topic) which became their assigned "group number."



Since we had three courses being taught in the Educational Laboratory Classroom, students were required to select the course subject.



After selecting the course a list of the enrolled students was populated with randomly generated emoticons ☺☺. A dialogue box confirmed your net id selection and if there was only one prepared document for the class the application loaded and the students got to work.



In the event that the instructor had prepared multiple worksheets (lecture documents) for a class session, students were prompted to select their “assigned” document. Figure 7.5 shows an example of a class session where students were assigned either the “blue” or “yellow” worksheet.

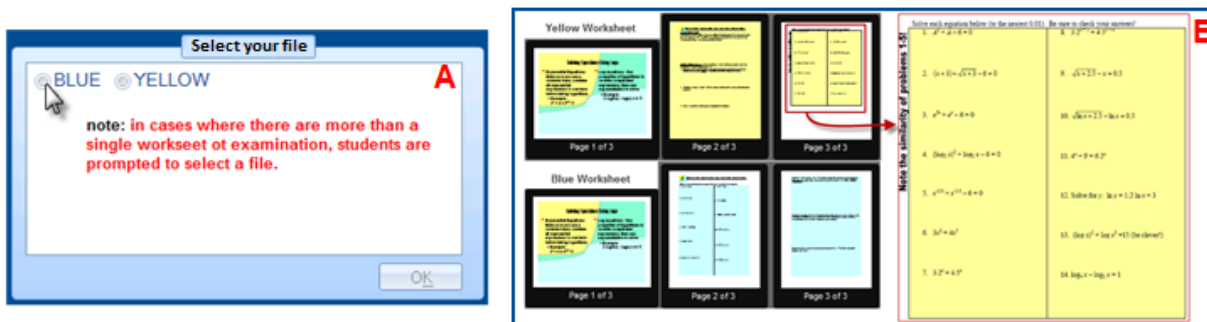


Figure 7.5: A) On days with multiple documents deployed students selected their assigned worksheet. B) “blue” and “yellow” documents available for selection by students in the previous dialogue

In the application interface of the NuPaper Classroom® Study, we provided a minimal interface stripping away a number of extraneous tools that may have encouraged students to “toy around” with the system (such as the “shape recognition tool”, and the ability to insert and delete pages, to name a few). We reduced the user interface even more when students were in “examination mode.” While in examination mode students were limited to a black inked pen and stroke erasure.¹ Undo and Redo functionalities were also available (figure 7.6).

7.5.2 Data collection

Key to the study was the ability to record and display students’ interaction with the Tablet PC computers. The finest granularity of information recorded by the pen-input devices is a collection of ink strokes. Each ink stroke contains pressure information, X and Y coordinate location of individual points making up the stroke, time of stroke creation (when the stroke was first started), duration of stroke creation (how long the user was inking), and other attributes such as color and id of ink stroke (figure 7.7). (This is similar to figure 5.22 on page 123).

By aggregating ink strokes for each page (multiple pages make a document), we reported summary information for each page such as ink density—how much ink in some human understandable metric—the number of strokes, average speed of stroke, number of strokes deleted (erasures), and time aggregates including spectator time (no inking—observation) and inking time (writing time). By creating models of interaction such as annotating prepared instructor slides or performing assessments, our system provided more meaningful representations (display) of a classroom’s state. Measuring these attributes allowed us to test our hypothesis that a dashboard display could increase a teacher’s understanding of her students’ level of engagement and that this information could be useful. We also compared a number of different dashboard

¹The difference between the *stroke* and *point* erasure is that the stroke erasure removed ink strokes (which are made up of a number of points) while the point erasure simulated the traditional erasure on a pencil.

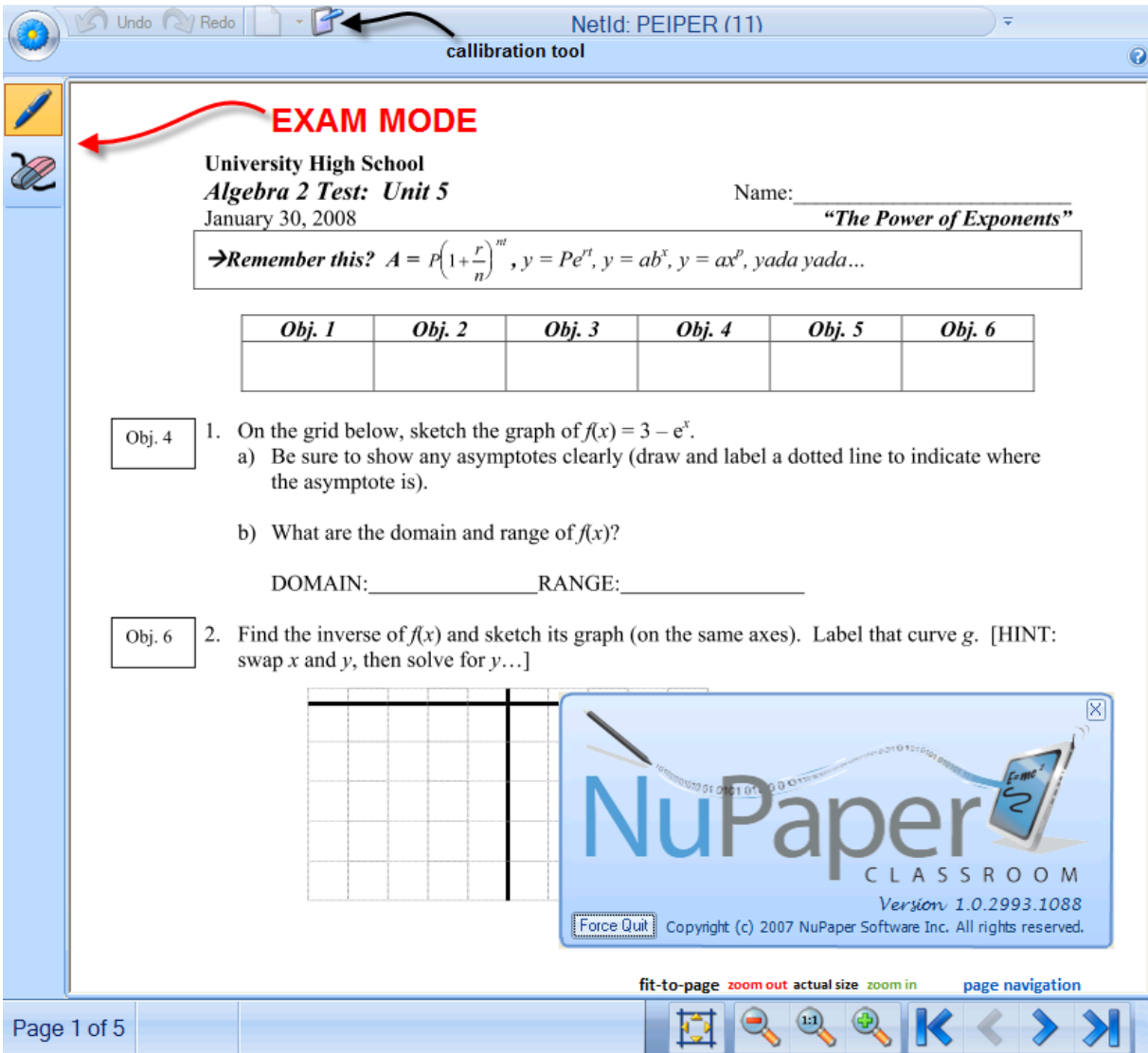


Figure 7.6: Illustration of the NuPaper Classroom® examination interface


```

22:23:52.807. Captured an Ink Stroke
ID = 298
Color = ff5e9200 where this represents the hex value of the ink color attribute
Bounding Box = {X=1928,Y=6086,Width=592,Height=1032}
Points (37): Points (37): {X=2489,Y=6163} {X=2433,Y=6134} {X=2364,Y=6118} {X=2291,Y=6104} {X=2215,Y=6107} {X=2144,Y=6112}
{X=2064,Y=6128} {X=2010,Y=6148} {X=1970,Y=6174} {X=1949,Y=6202} {X=1949,Y=6234} {X=1973,Y=6264} {X=2017,Y=6299}
{X=2082,Y=6330} {X=2175,Y=6370} {X=2255,Y=6415} {X=2337,Y=6465} {X=2408,Y=6522} {X=2460,Y=6585} {X=2492,Y=6655}
{X=2498,Y=6725} {X=2481,Y=6801} {X=2445,Y=6872} {X=2397,Y=6936} {X=2341,Y=6992} {X=2285,Y=7039} {X=2234,Y=7071}
{X=2186,Y=7092} {X=2150,Y=7096} {X=2117,Y=7092} {X=2096,Y=7066} {X=2070,Y=7050} {X=2058,Y=7019} {X=2044,Y=6989}
{X=2043,Y=6953} {X=2047,Y=6915} {X=2067,Y=6872}
Packet Descriptions:
Normal Pressure (37): 70 111 151 193 205 214 220 223 226 228 229 230 231 231 232 233 233 233 234 234 235 235 236 236 237 237
237 236 234 231 223 203 161 119 76 34 0
Timer Tick (37): 0 11 11 21 31 41 41 51 61 71 71 81 91 101 101 111 121 131 131 141 151 161 161 171 181 191 191 201 211 221 231
231 241 251 261 261 271

```

Figure 7.7: Example of an ink stroke’s attributes recorded in the log files

designs. Dashboards convey information implicitly (unobtrusively) obtained from the students’ interaction with their Tablet PC. For purposes of this research, implicit information was considered to be separate from information obtained explicitly (obtrusively) by the teacher probing the students verbally or through the computer. That is to say, obtrusive measures include any information explicitly pushed to a student’s device for the sole purpose of polling the students during an already established activity.

7.6 Dashboard Design

In this section we provide an overview of the four dimensions of dashboard content: *summary*, *selection*, *time interval*, and *profile*. Summary refers to a display of either a student’s actual work, or some quantitative synopsis information. Selection refers to the number of students whose data is displayed—that is, whether the class as a whole is displayed or just a subset of the students. Time interval refers to the period of time or frequency that the data are aggregated and/or presented. For example we could display a representation of the classroom activity measured within the first five minutes of a single lecture or display a summarization of activity recorded in the first six weeks of the semester. Profile relates to whether the students’ prior behavior is taken into account (i.e. whether the system maintains a historical “profile” of the class) and used to modulate the dashboard display. In addition to these dimensions, there is always the overriding issue of data visualization—given a particular choice of data points to select, there are numerous ways to present them—and of control—how the teacher may influence the dashboard display during class.

We elaborate on the four dimensions listed above.

7.6.1 Summary: actual work vs. derived data

Showing the students' actual work means displaying the students' whiteboard contents, thus providing the same kind of information that a teacher would get by walking around the classroom. Depending upon the selection dimension, this might involve a series of thumbnail views of students' whiteboards (which, depending on the size, could be hard to read in detail, but might still be useful), or a single screenshot (with a mechanism for switching to different students), or a small set of larger thumbnails. The advantage of showing actual work is the detailed information it provides, but the disadvantages are that it may be too much information for the teacher to absorb in real time, and that it may be so cluttered that the teacher can, in reality, obtain no useful information. The term "screenshot" is somewhat misleading since it implies creating an image of the whiteboard contents. Although thumbnails (defined below) are almost always scaled-down versions of bitmap images, our student "thumbnail displays" (as presented in section 7.7) were actually replica student whiteboards. Instead of taking screen snapshots at some regular interval of the student Tablet PCs and updating the dashboard display (as is typical of the VNC protocol), we recorded all student interactions, and sent representations of the interactions to the dashboard. That is to say, our student thumbnail displays were actually mini whiteboards. As a result, the display of student work was no longer limited to pictorial representations of their current whiteboard state. Instead, our thumbnail displays were of higher fidelity and much easier to read than the typical scaled image—since the scaling factor did not degrade the "picture" quality.

Thumbnails have been integrated into web-based photo albums by web designers in order to reduce download time of pages. By displaying smaller-sized images (geometrically and by file size) users can more quickly navigate through images. Similarly, operating systems and desktop environments have integrated thumbnail views into their file explorers. Standard document types such as TEXT, PDF, PNG (to name but a few), provide image representations of their files. Figure 7.8 shows a directory of images in "view by medium icons" mode in Windows Vista. The directory contains eight images in PNG² format. Although it is difficult to discern what the content of the file labeled "EX2" contains, by increasing the size of the icon from medium, to extra large, the thumbnail representation becomes more readable. The extra large-sized thumbnail in figure 7.8 is approximately 45% of the original image size.

Showing derived data means drawing graphs of physical characteristics, such as amount and speed of writing. The advantages of this approach are (1) it requires much less space per student, so we can have a more readable display of more students; (2) by using numerical data, it allows for comparisons that may be useful, such as quantities of ink during different periods of the class; (3) potentially, the computer may be

²PNG stands for Portable Network Graphics and is a bitmap image format that employs lossless data compression (typically uses the "png" file-extension). source: <http://en.wikipedia.org/wiki/Png>

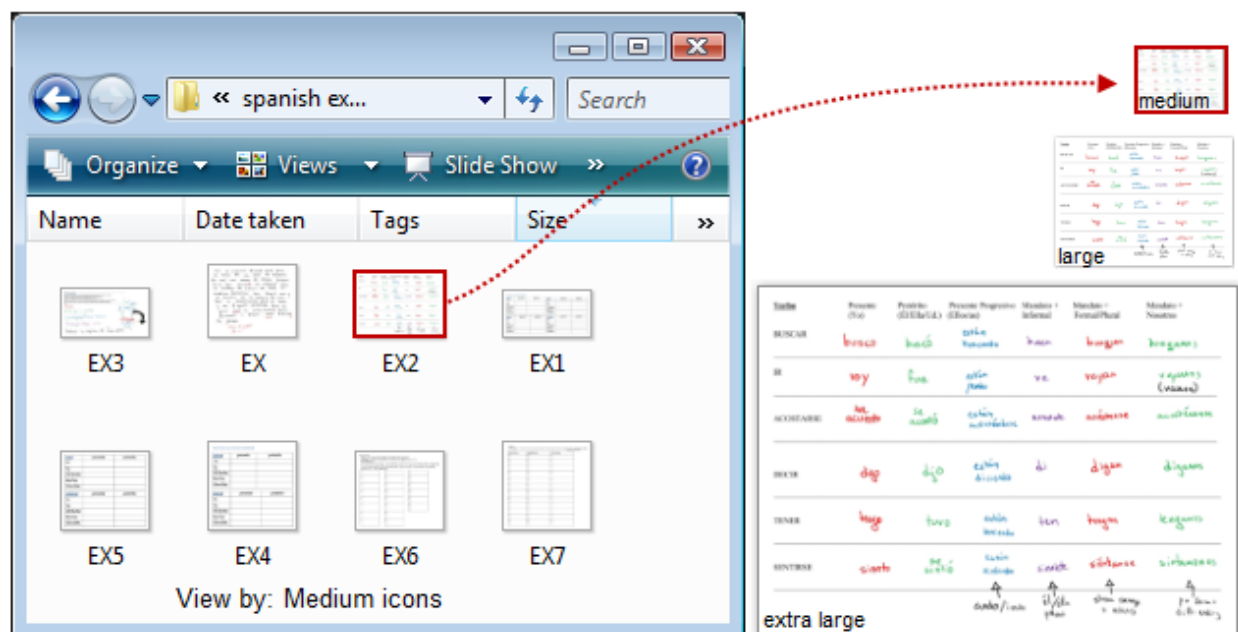


Figure 7.8: Thumbnails in the file explorer of Windows Vista

able to deduce information that the teacher cannot easily infer (much less, in real-time for many students). For example, an instructor might be inclined to meet with a student who failed to answer a question on the exam if it was brought to his attention that the student spent the majority of her time on this particular question.

7.6.2 Selection: entire class vs. selected students

Showing all students gives a general picture of the class, which the teacher might find useful as a way of determining whether the teaching level is appropriate: is she going too fast, leaving students behind, or going too slow, or going just about right? In effect, this allows the teacher to judge herself, using objective feedback not available in the traditional classroom. Showing individual students allows for more detail about those students; this can allow the teacher to see how those students are doing. This is not entirely orthogonal to the other dimensions. In particular, it interacts strongly with the *summary* dimension, since the number of students displayed determines the amount of detail available per student. It also strongly implicates the user interface (i.e. the teacher's control of the display), because a display of selected students may be useful only if there is a convenient way to change the selection in real time.

7.6.3 Time interval: data aggregation period

The “time interval” dimension is exclusively used when providing summary display information. Do we provide a “picture” (some method of summarizing the activity in the classroom) of the class right now or over the past so many minutes? Summary information necessarily involves a non-zero length time interval. (It makes no sense to speak of, for example, “instantaneous inking volume.”) But that interval could be 30 seconds or 15 minutes. (Detail views of student whiteboards would have to be updated in real-time because they occupy too much screen real-estate to include several visible whiteboard representations per student.) The advantage of a short time scale is that it provides an instructor with information of what is happening now. Since it contains less data, the data presented can be larger and more visible. The advantage of longer time scale is that it may allow teachers to see how his or her students have progressed during a lesson or how they responded to various topics during a class. Over a longer time scale, the teacher might also be able to identify trends of student behavior.

7.6.4 Profile: the use of historical data for comparisons

The display may take into account the historical behavior of individual students or of the entire class. For example, a student’s inking behavior might be average, but that would be noteworthy if that student usually is a prolific note-taker. Similarly, just as the teacher might be interested to know that students are writing differently now from fifteen minutes ago, she might like to know if they’re writing differently from how they did in the previous class or a class several weeks ago.

Choosing a point along each of these dimensions determines what data will be shown on the dashboard. But the visualization of that information can vary in many ways, affecting the density and readability (in real-time, which is different from book-time) of the dashboard. Nor is there any reason why different regions of the dashboard might not have different data displayed. And the display may change during the class, for any number of reasons; in particular, the display may change as the class changes “mode,” e.g. shifts from ordinary note-taking to a student exercise; or, the teacher may be able to request a change, bringing user interface issues into play.

Thus, the design space is enormous. However, we wanted to address one question: whether a dashboard—any dashboard—can help teachers teach. Our hypothesis is that a dashboard, carefully designed, can increase the teacher’s knowledge of what the students are experiencing, and that this knowledge will be useful.



Figure 7.9: Dashboard display spans two monitors A) (1280x1024) and B) (1600x1200) and a Wacom (1280x1024) tablet C) serving as the 3rd monitor, and providing instructor pen-input to individual student workspaces. The separation between the two upper monitors (A and B) was less than ideal. The instructor also used a Tablet PC connected to the room's projector.

7.7 New Design Approaches

All design involves trade-offs and the navigation of a design space. What is of interest in this case is the size and complexity of the design space and the nature of the trade-offs. As an illustration we elaborate below on a single trade-off—just one of many that we considered. The thumbnails are easy to understand, and to integrate into the existing practice of teachers. They look like paper documents. Teachers are already skilled at glancing at students' paper worksheets, often from considerable distance and frequently when the paper is upside down from their perspective, and yet still gaining useful impressionistic data: “Has he done anything at all? Is she just doodling? Does that look like it might possibly be a reasonable answer?” Unfortunately even thumbnails take up considerable space, particularly when there are, say, 24 of them.

As a result we investigated a number of large-screen display options. We first tried to combine the LCDs already available in our laboratory with a nineteen inch LCD with a maximum resolution of 1600x1200 pixels (figure 7.9). The disparity between the resolutions of the two LCD displays was problematic when trying to expand the dashboard across the two displays. When we obtained another nineteen inch LCD (figure 7.10) we integrated the two monitors (both with the same maximum resolution of 1600x1200) in creating our display. Our final “dashboard display” combined five monitors, four of which were displaying at a resolution

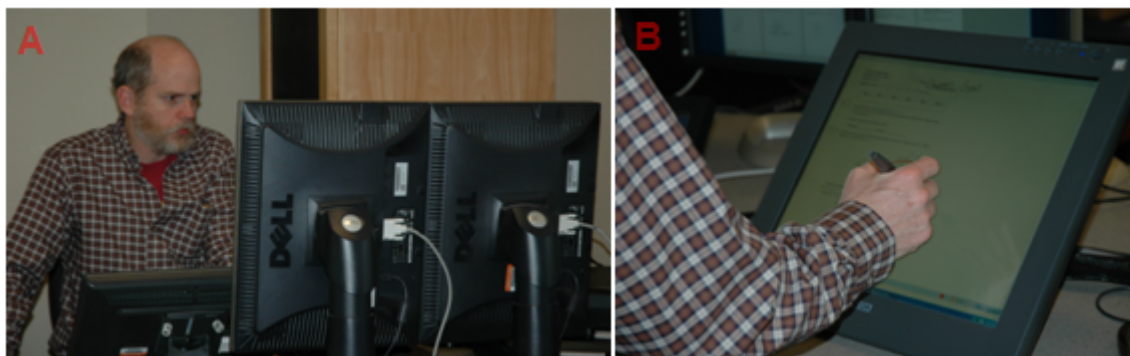


Figure 7.10: A) Front view of instructor using the two-monitor dashboard display B) Example of instructor using the WACOM device for pen-input interaction with student work spaces

of 1200x1600 pixels (rotated), and a fifth Wacom pen-enabled (1280x1024) display, allowing the instructor to annotate and interact with a student's whiteboard workspace (figure 7.11).

Given the continual decline in the cost of monitors, this does not seem an infeasible setup to for a classroom that would also have 24 networked tablet PCs for students. However, even with this more generous than typical hardware setup, we remain pixel-constrained. Figure 7.12 illustrates the latest version of the visualization which is clearly designed to exploit every available pixel.

In the current version, each thumbnail is identified by a small amount of text (the student's login ID), a number indicating which group they are in and that student's current page (i.e. 2/4 meaning page two of four). In live observations of use we have found that teachers refer to the overview of thumbnails, looking at the display for just a few seconds before deciding which individual or group to attend to next. Although there are various built in options to sort the thumbnails based on derived data (such as amount of inking), these options are used rarely (figure 7.13) except for the sorting by group number feature which allows the instructor to quickly glance at the work spaces of students within the same collaborative group.

The facility to zoom in on a particular thumbnail of interest is used heavily (figure 7.14). We have taken these observations as informing subsequent design thinking where speed of interpretation and ease of integration with other information (conversation fragments, noise level, body position, etc.) take priority over more detailed visualization refinement interactions by the instructor. This can be a difficult decision to make from the perspective of traditional information visualization. It is tempting to add yet more options and functionality to provide the teacher with more choices, better alternate views, charts, rankings etc. But in the real-time context of in-class activities, speed, simplicity and ease of integration into conventional pedagogy outweigh more advanced options.

In future work we will investigate arranging the thumbnails to more closely resemble the physical classroom layout, with students working in groups around tables. With a more faithful classroom-plan layout,



Figure 7.11: Dashboard display spanning 4 monitors rotated at 90° and a Wacom tablet serving as the 5th monitor providing instructor pen-input (see part 'B' of figure 7.10) to individual student workspaces.

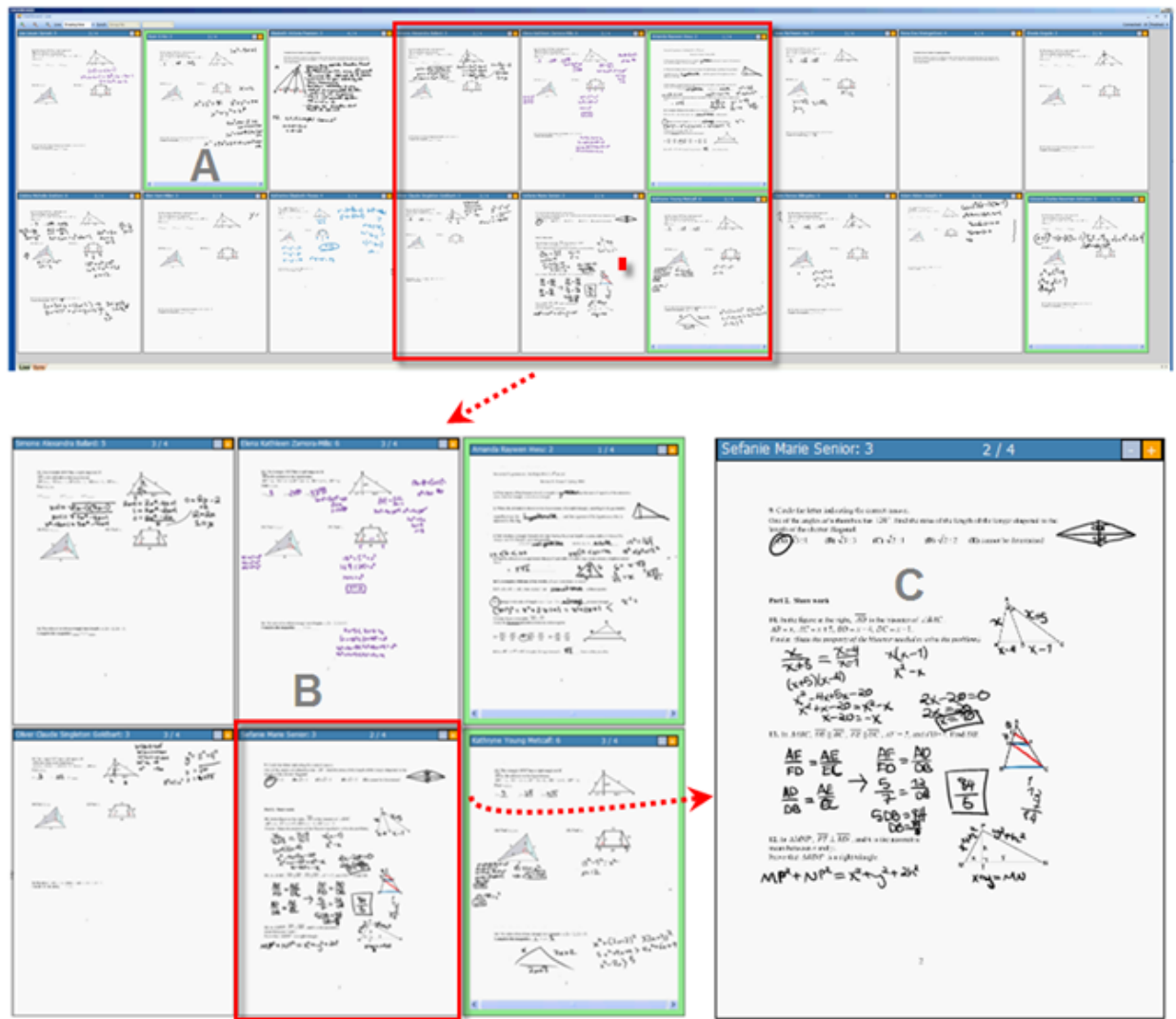


Figure 7.12: A) The dashboard display spanned across four monitors 4800x1600 B) Magnification of the middle six student workspace thumbnail displays C) magnification of a single student workspace thumbnail

Dashboard - Live

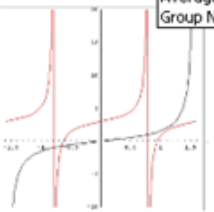
Live **Synch**

Sophia O Shenk: 1

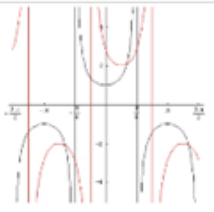
3/28/2008

Finishing Off Reciprocal

The black graph in each picture has undergone one vertical transformation. For each picture (with specific numerical values), write down the transformation.



Black graph: $y = \tan x$
Red graph: $y = \tan(2x) + 3$



Black graph: $y = \sec x$
Red graph: $y = 2 \sec(x - \frac{\pi}{4})$

netID

current page

strokes created

strokes deleted

Ink Length

Writing time

Erasing time

Average pressure

Average speed

Group No

Taka Otsuka: 3

3/28/2008

Inverses as Functions

Complete this table (inverses of sine, cosine, and tangent) ... recall:

- How are the domain and range of a function related to the domain and range of its inverse?
- When we began discussing inverses of these functions, we discussed "half angle" restrictions... where will those show up in this table?

(Inverse of)	Function	Domain	Range
Sine	Arccsin	$[-1, 1]$	$[-\pi/2, \pi/2]$ (right half)
Cosine	Arccos	$[-1, 1]$	$[0, \pi]$ (left half)
Tangent	Arctan	all real x 's	all except odd multiples of $\pi/2$ (right half)

How are the graphs of a function and its inverse related (there's a geometric answer)?

On each graph below, plot the inverse functions (keep "range" in mind):

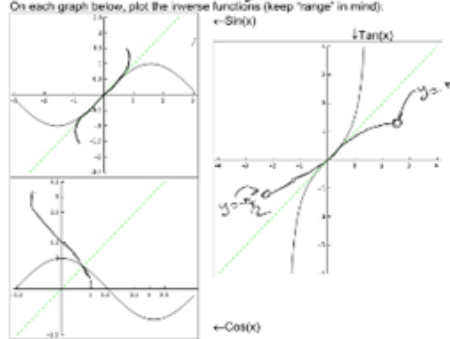


Figure 7.13: Example of synchronous sorting options for student workstations on the dashboard display; net id, current page, strokes created, strokes deleted, ink length (total amount of ink in pixel space), writing time, erasing time, average pressure, average speed, and group number.

grsmith1 - Sync

- all numbers "00" are not final.
- each number is on its own line.
- The radius of the cylinder is 5.7 cm and its height is 3.7 cm.
- Volume of a cylinder is $V = \pi r^2 h$.
- Volume of a cylinder is $V = \pi r^2 h$.

Write an expression for the volume of the cylinder in terms of the radius of the cylinder and the height of the cylinder. Write your answer in terms of π , r , and h . Write your answer in terms of π , r , and h .

directions: you should be writing a ratio, not an equation

$$49\left(\frac{4}{3}\pi x^3\right) = \pi(5.7x)^2 3.7$$

$$\frac{4}{3}\pi x^3 = \frac{\pi(5.7x)^2 3.7}{49}$$

$$\pi x^3 = \frac{4}{3} \left(\frac{\pi(5.7x)^2 3.7}{49} \right)$$

$$x^3 = \frac{4}{3} \left(\frac{\pi(5.7x)^2 3.7}{49} \right)$$

$$x = \sqrt[3]{\frac{4}{3} \left(\frac{\pi(5.7x)^2 3.7}{49} \right)}$$

x = .32280612244898

Directions: Simplify + round to nearest 0.01

$$\frac{(\frac{4}{3})\pi x^3}{\pi(5.7x)^2 3.7} = \frac{(\frac{4}{3})x^3}{(5.7x)^2 3.7}$$

THINGS HAPPEN

Viewing page: 1 of 4 Student page: 3 of 4

Live Sync

Figure 7.14: By double clicking on a student's thumbnail on the dashboard (see figure 7.13) the instructor gets an enlarged screen of the student's whiteboard with facilities to navigate through the student's document and annotate directly on the student's workspace.

mapping from thumbnail to student would be much easier for the teacher, but the tradeoffs are the loss of pixels to enable such arrangement and the consequent reduction in the size of the thumbnails, making them less easy to read and interpret. We will need to study the system in use to understand if this trade-off is worthwhile, and indeed what are the hidden factors in the trade-off that can inspire future design innovation.

7.8 Evaluation

The fall 2007 dashboard design was tested in a number of class settings: in a first year university level Computer Science course (presented in section 7.8.1), and in four classes at the neighboring University of Illinois Laboratory High School including courses in accelerated Spanish³, Algebra 1, Algebra 2, and Geometry. The algebra 1 instructor, Mr. Gene Bild, taught one of his three sections of algebra in our educational laboratory classroom for the duration of the fall semester. The second course offered in the fall semester was algebra 2 taught by Mr. Craig Russell. All three of Craig's classes had the opportunity to participate in the study. In the fall 2007 semester the 8:50AM and 2:05PM sections of algebra 2 were taught in our laboratory and in the spring 2008 semester, the 12:15PM and 2:05PM sections were taught in our laboratory. In the spring 2008 semester a section of geometry taught by Mrs. Ioana Boca participated in the study. The two sections of algebra 1 and geometry *not* participating in the pilot study were surveyed regarding their perception of their instructor's performance and teaching styles in the traditional setting in the University Laboratory High School. The complete survey and results for all the sections can be seen in appendix A.3.

In all but question 17, the possible answers of the survey questions were 1 (not at all), 2 (rarely), 3 (sometimes), 4 (often), and 5 (almost all the time); thus, higher-numbered responses are better. Questions of interest were questions six through nine which attempted to show if a noticeable difference in the instructor's cognizance of their students was in fact perceived by the students.

Q6 How often do you feel the instructor offers you one on one attention during class?

Q7 How often do you feel the instructor is able to identify that you were struggling with a particular topic?

Q8 How often do you feel the instructor is able to identify that you understand the material?

Q9 How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

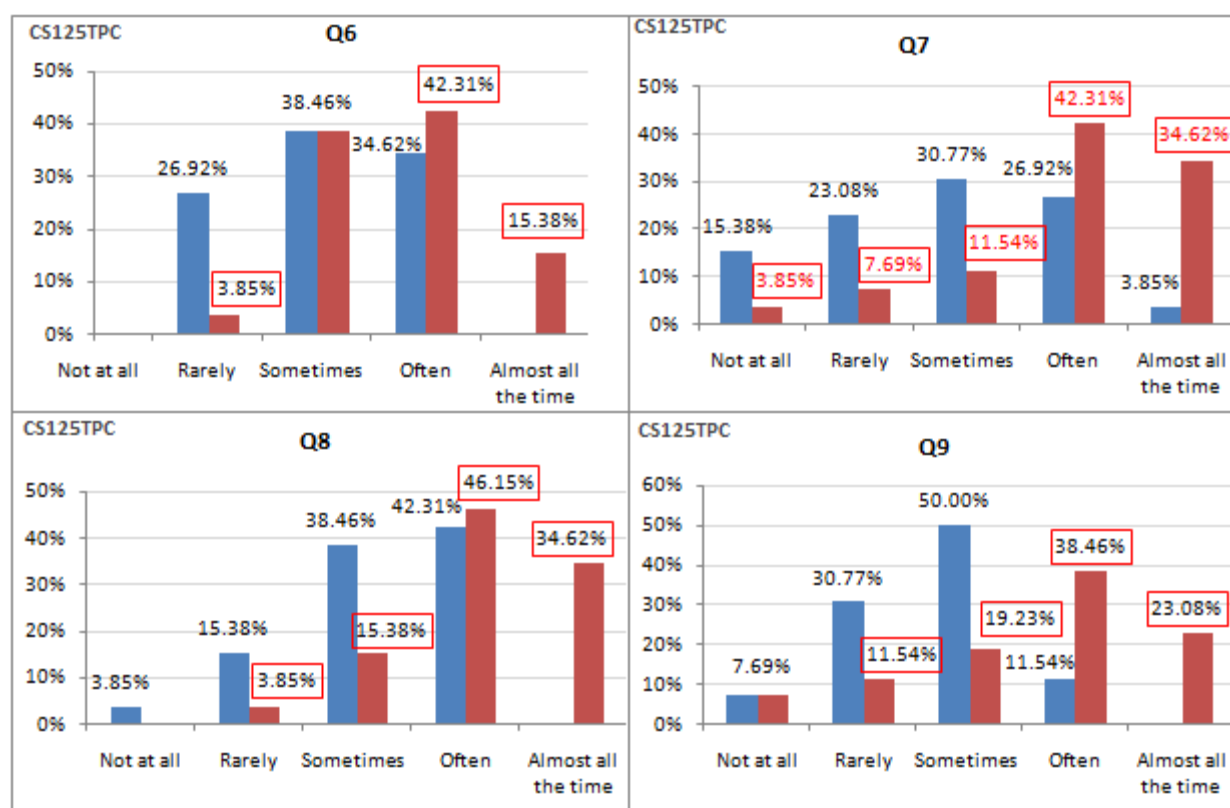
We continued the study in the spring 2008 semester and present a summary of the survey evaluations below. These studies involved relatively skilled teachers and motivated students, so the initial results need to be treated cautiously. We were interested in identifying problems that the teachers had in using the technology

³We did not survey the Spanish class since they did not use the Tablet PCs on a regular basis.

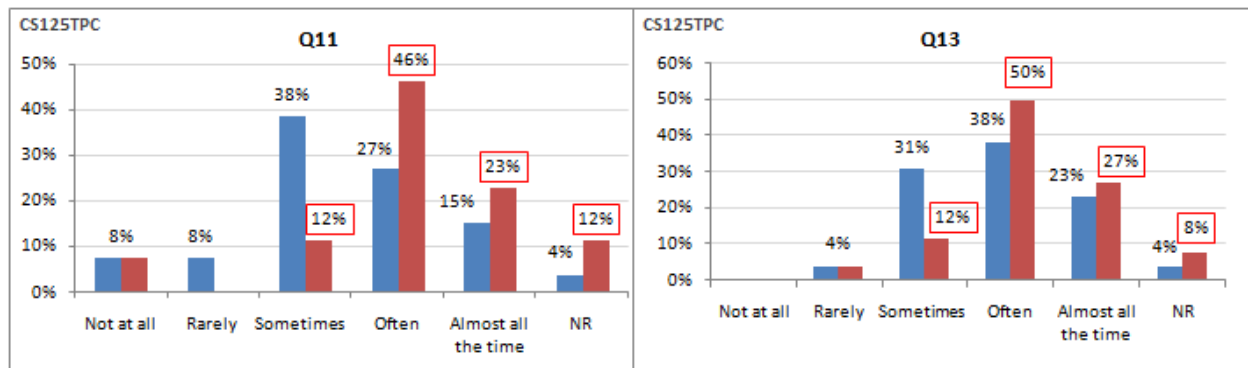
and ways that they integrated it (or failed to integrate it) into their existing teaching practices. The problems in situated use, as well as the unexpected appropriations of the technology for unexpected purposes were used to inform subsequent rounds of prototyping. In section 7.8.1 we present the results from the CS125 tablet-section pilot study. In sections 7.8.2 and 7.8.3 we present the results for the algebra 1 and 2 courses and conclude with the results of the Geometry course in section 7.8.4.

7.8.1 CS125 TPC survey results

The results from the university level Computer Science were extremely positive. Questions six through nine show a clear increase in the instructor's attentiveness and level of awareness.

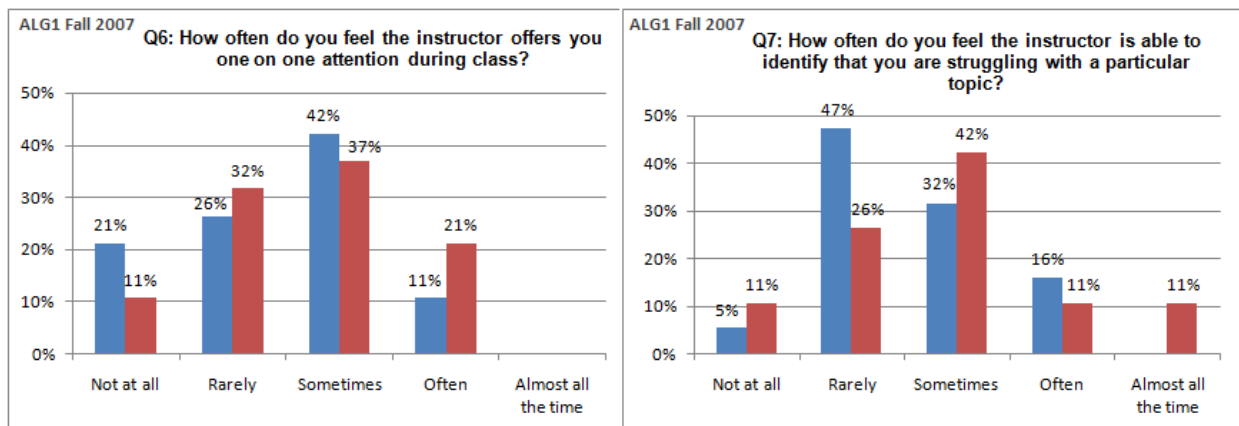


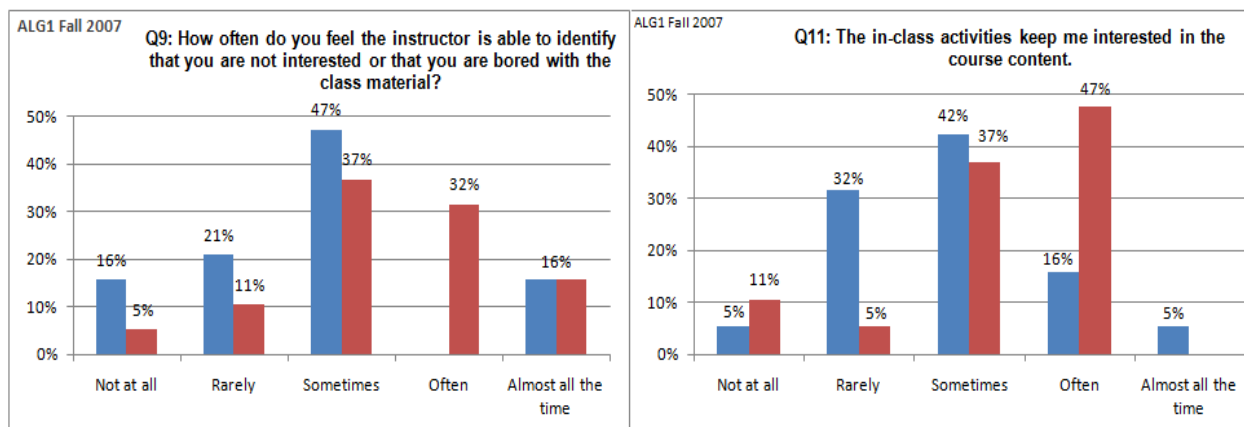
Questions 11 (in-class activities keep me interested) and 13 (there is a positive interaction between students and instructor) show similar improvements.



7.8.2 Algebra 1 survey results

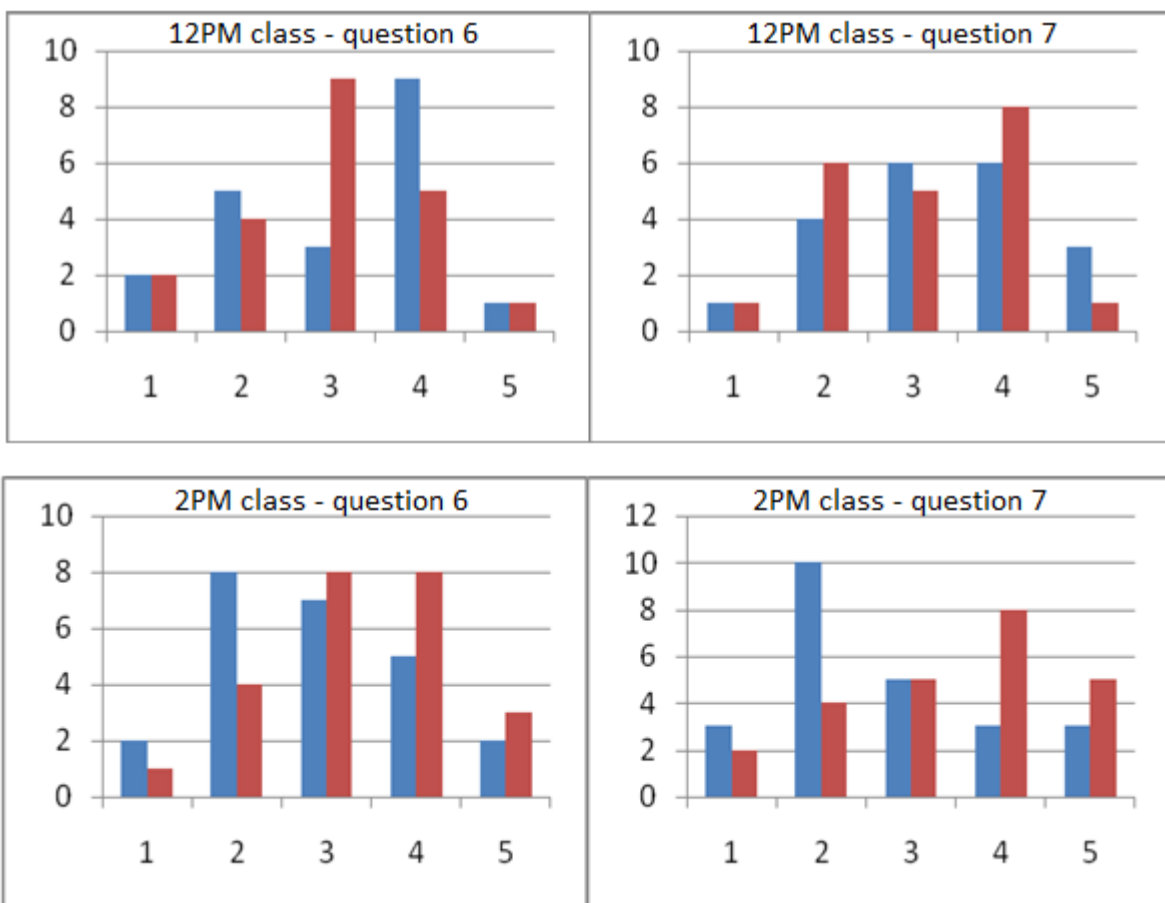
There was a total of 60 students enrolled in the three algebra 1 classes. Thirty-eight of these students did not participate in the pilot study, but they were surveyed. All of the results are presented in appendix A.3. For the 22 students that did participate in the study, we administered the survey in December, and asked them to answer the questions twice, once as they felt in December, and once as they would have answered in September. Questions six, seven, nine, and eleven show substantial changes in the instructor's attentiveness.





7.8.3 Algebra 2 survey results

In each of the algebra 2 classes we hosted, the students spent some time being taught by Mr. Russell before moving to the laboratory classroom. The fall classes were in their regular classrooms for about a month before moving to our classroom; the noon class, which used our classroom only in the spring, was the continuation of a class that met in a regular classroom in the fall. Thus, all students had some experience being taught by Craig Russell both with and without the dashboard. We present a subset of the evaluation results for the fall 2007 section of the 2PM class, and the noon class in the spring. (The complete survey results can be seen in appendix A.3. There were a total of 66 students enrolled in the three algebra 2 classes.) The circumstances of the two surveys were very different. For the 2PM class, we administered the survey in December, and asked them to answer the questions twice, once as they felt in December, and once as they would have answered in September. For the noon class, we administered the survey twice, when they first started using the Tablet PCs (again, this is after a semester with Mr. Russell in a conventional classroom) and again in May. The following graphs present the count of before-and-after responses; the bars on the left (in blue) are “before” responses, and bars on the right (red) are “after” responses.

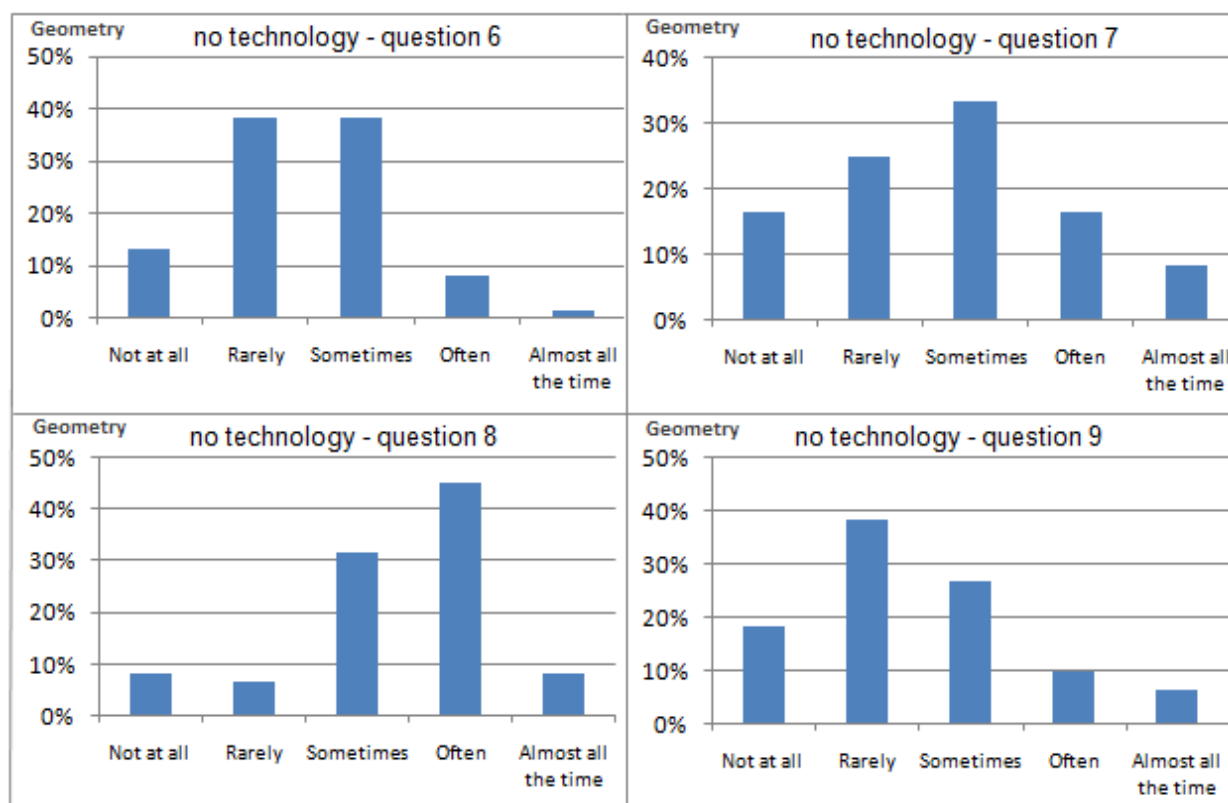


The 2PM class shows a strong effect, while the noon class's responses are more mixed. This is most likely due to the different circumstances in which the surveys were administered. It is not clear to us which numbers are more believable. The true before-and-after surveys, as were done in the noon class, seem more objective, but that is not necessarily true; the problem with surveys administered four months apart is that the students become used to a certain level of attentiveness from the teacher, and their answers reflect this; in other words, their definition of the term “often” changes. The 2PM surveys offered students the opportunity to provide a head-to-head comparison of two classroom situations—with the dashboard and without; their responses indicate that they felt a definite change in their teacher's ability to respond to their problems quickly, and their feeling on this is surely meaningful.

7.8.4 Geometry survey results

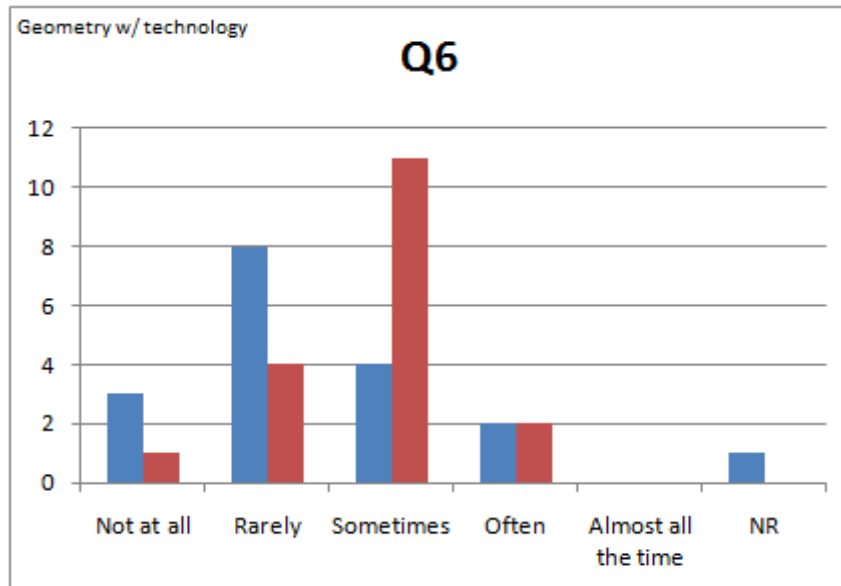
There were 60 total students enrolled in three sections of geometry, of which 18 students were in the 1:05PM class participating in the study. The survey results for questions six through nine for all 60 students are

displayed below⁴.

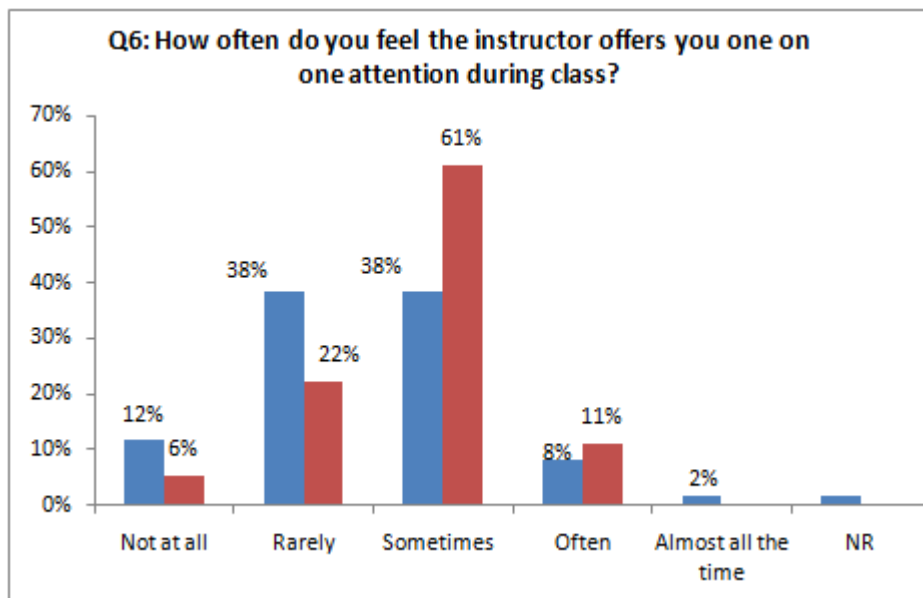


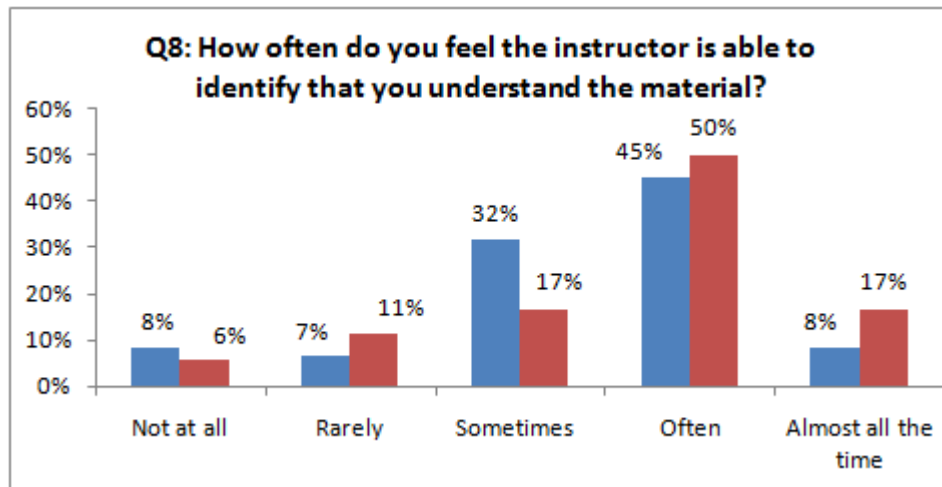
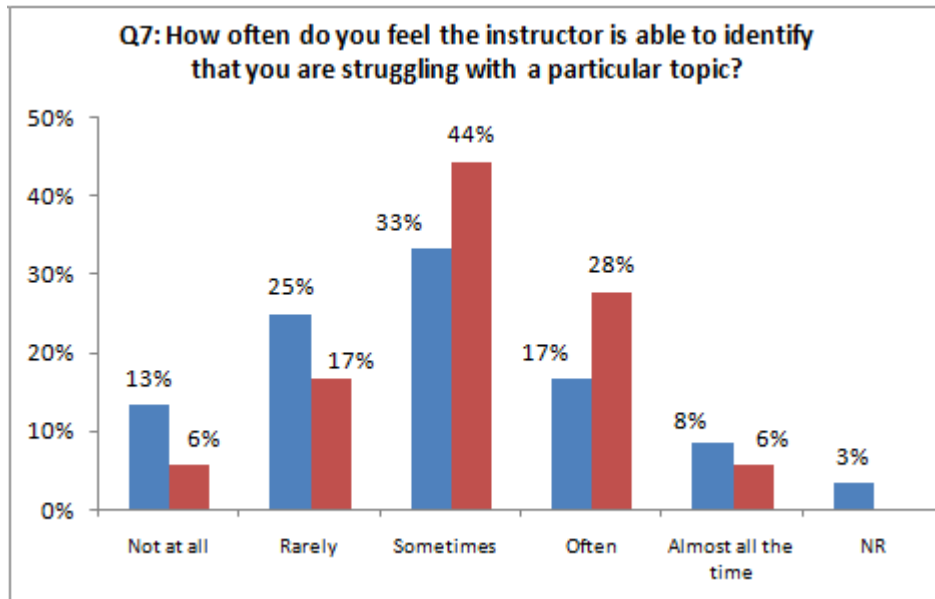
With the exception of question eight, the student evaluations of the instructor’s attentiveness to her students was rated “rarely” to “sometimes.” The results of question six for the students who participated in the survey illustrate a clear improvement after the integration of the dashboard into the classroom.

⁴The “no-technology” survey includes the initial survey responses from the 18 students who participated in the pilot study. We surveyed the class prior to participating in the pilot study.



When we looked at comparing the measurement of all of the students prior to the study with the exit survey results of the students who participated in the study, an improvement in the instructor performance was verified.





7.9 The Larger Use Design Space

As noted, the in-class use context creates a dauntingly large design space to explore for developing useful, usable and acceptable visualizations of state and relative progress. But this is just a subpart of the even larger design space of uses to which the visualizations of the captured data can be put. These include:

- Reflection by the teacher on what just happened after the class has ended
- Consideration by the teacher of sequences of classes, even of a whole semester of teaching a class
- Comparison between activity in different classes and class sections
- The teacher trying out small innovations in pedagogy around using tablet PCs and reflecting on the results

All these involve use just by the teacher herself acting as a reflective practitioner [158]. There are also uses involving others:

- Expert-teacher master classes for sharing pedagogical innovations
- Showing a class interaction at very high speed—like a speeded up action replay, with commentary explaining what was being done and why, problems arising and how they were resolved
- Mentoring of trainee teachers such as going over a lesson and noting strengths and weaknesses
- Continuing professional development
- Conventional educational research into classroom activity, performing highly detailed (and typically laborious analysis) using the data obtained and visualized from Tablet use as an alternative to or as a supplement to more conventional data capture techniques such as videotapes of the class
- Detailed evaluation of tablet PC technologies, mechanisms, teaching tools and pedagogies. That is, these points in the visualization design space can serve as resources for the iterative design and testing that occurs in exploring the pedagogical design space.

What remains to be discovered is which features of the visualization are common to all (or most) of these and which are unique or have different relative importance. The underlying data is likely to remain the same. This is determined by what is possible to collect, and one might as well collect all that can be collected, given that the overheads of collection and storage are minimal. The difficulty lies in the variations in the trade-offs of different ways to process this data and how to show it. Much depends on the amount of time available to devote to analysis in each of the above contexts. This can be considered as the ratio of class time to analysis time. So for a 50 minute class, (or a one minute episode) how many minutes to devote to analysis? As an example, for video analysis, this ratio can be as much as 10:1 [3].

The setting of in-class exercises is at one extreme of this continuum, where the amount of time for analysis should be minimal. Traditional educational research using video and conversation analysis is at the other extreme, with the other cases residing somewhere in between, highly dependent on context and indeed on what the technology can be designed to afford.

7.10 Conclusion

Tablet PCs create both an opportunity and a challenge. This applies to issues of how to use them in teaching and in the technologies developed to support this use. One such issue is how to go about processing the unprecedented amount of data that is obtainable describing a student's interactions with the Tablet PC. One challenge is providing evidence of a student's learning and overall cognitive state (engaged, confused, bored, frustrated etc.) during class. This opens up potential for improving overall teaching effectiveness by more individualized help in a classroom context. Another challenge is that this data only provides indirect

evidence about students' cognitive states. It needs to be processed and in particular it needs to be visualized and integrated with the other sources of evidence available to teachers.

This chapter has outlined our early work in exploring this intriguing design space through iterative prototyping and evaluation in classroom settings. Our focus has been on at-a-glance information that is collected and aggregated from student machines in real-time. We have shown how this can be done, but additional research is required to help design and develop intuitive visualizations for real-time classroom usage. The user interface design required some level of interaction with the dashboard system in order to select how to sort/view the student workspaces. Over the course our study, we observed that the system required interaction that hindered the ability of some of the instructors, requiring small amounts of supplementary information to help them re-establish a sense of overall classroom context after interacting with an individual.

Chapter 8

Conclusion

In this thesis, we have presented a number of studies on the use of Tablet PCs in the classroom. All point to the potential for Tablet PCs to enhance education and educational research. Tablet PCs can be used to record classes in great detail for research purposes; they can be used to make the class more “communicative,” thereby increasing student engagement and reducing student frustration; and they can be used to provide non-obtrusive monitoring of students, providing greater transparency for the teacher.

Clearly, none of these studies—more any of the prior studies cited in this thesis—are conclusive. With one exception—the summer, 2002, study—our studies failed to show actual improvements in learning, and this is typical of studies in this area. However, the results do show that teachers and students were generally appreciative of the technology, suggesting that it may have made learning more enjoyable and more efficient.

We certainly feel safe in concluding that this technology offers promise. There is a great deal of research yet to be done:

Technology. In many studies, including ours, technical problems prevent a clean test of the underlying hypothesis. In this field, details matter a great deal: A network delay of just a few seconds, or system crashes, even if rare, can sour the participants on the technology. A common complaint is that the Tablet PCs are too small—they are not even as big as a normal pad of paper, and students often work on several sheets of paper at once.

These are the main issue we see here:

Hardware form factor. Tablet PCs are a compromise: small enough to be reasonably portable, and large enough to write and read on comfortably. Which is more important in the educational domain: portability or screen real estate? There is some evidence that when people have their own Tablets, they use them more, which argues for portability, but, as noted above, in the classroom setting, screen size is paramount. Emphasizing screen size, we can imagine a class in which each student desktop is a large, pen-enabled computer screen—providing several times the writing area available on a Tablet. Emphasizing portability, we can imagine a smaller Tablet PC—say 5-by-7

inches—that is still large enough to use in class but lighter and easier to carry. Indeed, both form factors exist, but only large-scale studies can determine which is best for any given educational setting.

Software. Software that is to be used in classrooms must pass the most stringent reliability tests. At the same time, it must be flexible: the variety of subjects, age levels, and teaching and learning styles makes a one-size-fits-all solution untenable. Yet these two requirements are mutually contradictory: reliable software is expensive, which implies that only a small number of systems can be profitably built and marketed.

Classroom uses. This dissertation, and many of the references publications, point out a variety of ways in which Tablet PCs can be used to better engage students. However, these studies rarely disentangle the new affordances provided by a particular hardware and software configuration. In addition to devising new uses, rigorous studies of specific features are needed if we are to determine what works for education and what does not.

The impact of age level. In most studies, the participants have been resident college students. This is the easiest group for researchers to study, if only because the researchers are themselves college teachers. However, this may not be the group most likely to benefit from new educational technology. Indeed, it is difficult to imagine our dashboard study (chapter 7) being done in a college class rather than a high school class, because high school classes are, as a rule, *much more interactive* than college classes. Even now, the vast majority of college classes follow the traditional “sage on the stage” approach. Grade schools are liable to be even more appropriate settings for technology use. At the opposite end of the spectrum, adult education is often done online, where the Tablet PC technology might again be particularly useful.

Subject-specific tools. Almost all the Tablet PC tools described in the literature are generic; they are intended to allow a teacher of any subject—so long as the predominant teaching method involves a teacher and students sitting together in a classroom—to better engage the students and better “see” the class. The WIPTE workshops (<http://www.wipte.org>) always have some papers by teachers—mainly college teachers—discussing the use of Tablet PCs for teaching specific subjects. However, they usually employ generic tools. Again, the cost of software development is a huge barrier here. Still, one can imagine that more targeted software — for specific topics, or for students with specialized needs—could be tremendously useful.

Teacher’s dashboard. We have only scratched the surface of what might be done in this area. We noted in

chapter 7 how vast the design space of dashboards is. Here we focus on two broad (and not unrelated) issues:

Intelligent dashboards. Our current dashboard provides the teacher with a view of each student's work, but attempts to analyze it only to the extent of allowing for sorting the display on various measures. With the goal always of providing useful information "at-a-glance" the dashboard might be more helpful if it performed some intelligent filtering. Would it be possible, for example, for the software to say whether a student is performing differently than would be expected from his or her past behavior? Such filtering could result in a dashboard that is less cluttered and easier to read.

Large classes. In our studies, even with a very good teacher — one who knows his students well — and a fairly small class, the dashboard made a difference. Still, in such a setting, the traditional method of walking around the classroom is a decent fall-back position. In large classes, teachers rarely even attempt this; and in online classes (which may also be large), it is impossible. In fact, one could argue that the reason teachers in large classes don't make their classes interactive is precisely because there is no efficient way to monitor students; similarly, one could argue that this is the same reason that online classes are rarely synchronous: with no possibility of effective interaction, why surrender the convenience of asynchronicity?

Tablet PCs could alter these realities.

However, the kind of dashboard we employed in Chapter 7 is liable to be *less* effective in a large class, because it would contain too much information. Would it be helpful to view a small, random sample of students? Could the dashboard analyze the data from the students, as suggested above, to create a really useful summary of the class?

In this field, the immediate need is for compelling demonstrations of the utility of Tablet PCs in classrooms. Such demonstrations will fuel the desire for more classroom applications of Tablets, which will in turn lend force to calls for research in the areas we have just outlined.

Appendix A

Student Surveys

A.1 Chapter 5 Survey Questions and Results

We have three classes (taken over the course of two semesters) of survey data where we asked students to fill out questionnaires regarding their use of instructor-created whiteboard notes and their perception of how well the technology enhanced (if at all) the learning environment. In the Spring 2007 we surveyed two classes at the conclusion of the semester; CS125–Introduction to Computer Science and CS225–Data Structures and Software Principles. In the fall 2007 semester we surveyed the CS225 course. In total there were 333 surveys completed. The results for all 24 questions are presented below. These questions were based on an eClass [88] student questionnaires.

1. Did you regularly attend lecture? A.1
2. The classroom lecture was more engaging or interesting to me as result of the use of technology in class and the availability of notes afterwards. A.2
3. Because captured lecture notes are available after class, I was able to better pay attention to the lecture. A.3
4. The lecturer used the e-Fuzion technology effectively in class. A.4
5. This type of course is well-suited to e-Fuzion technology. A.5
6. Briefly describe your note-taking practices in classes not using e-Fuzion technology. A.6
7. Have your note-taking practices in this class changed as a result of e-Fuzion? If yes, briefly describe the change. A.7
8. Have you ever accessed the class Web page? Yes No (if no, skip to next section) A.8
9. How often did you access the captured lecture notes from class Web page? A.9
10. Why did you access the captured lecture notes? Check all that apply. A.10
11. All things being otherwise equal, I would prefer to take a class that uses e-Fuzion technology over the same class that does not. A.11
12. e-Fuzion technology will encourage students to skip lectures. A.12
13. e-Fuzion made me less worried about missing class. A.13
14. I trust that captured lecture notes will be available after every class. A.14

15. Audio augmentation of the Web-based lecture notes would have increased their value to me. A.15
16. Video augmentation of the Web-based lecture notes would have increased their value to me. A.16
17. Printing captured lecture slides after class for later review was valuable to me. A.17
18. The value of captured lecture notes would be greatly enhanced if replay of the instructor's ink was possible. A.18
19. What features of e-Fuzion did you find most useful? A.19
20. What features inside the class did you find least useful, or even distracting? A.20
21. What would you most like improved or added to capabilities of e-Fuzion? A.21
22. If your class had prepared lectures, AND you could print them before class and then take notes on top of them during the lecture, would/did you take notes this way? A.22
23. The availability of prepared lectures (printed out) before class would make it unnecessary to access the captured notes after class. A.23
24. If you accessed the captured notes, were you more likely to print them out instead of view them online? A.24

Figure A.1: Did you regularly attend lecture?

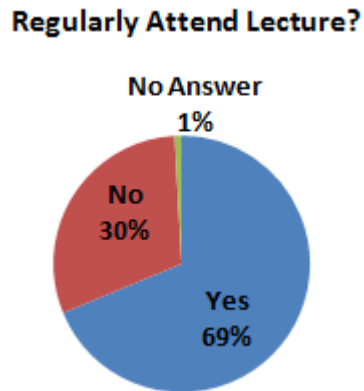


Figure A.2: The classroom lecture was more engaging or interesting to me as result of the use of technology in class and the availability of notes afterwards.

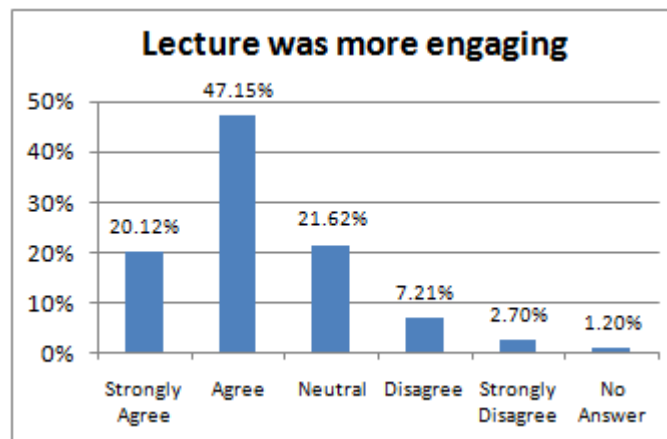


Figure A.3: Because captured lecture notes are available after class, I was able to better pay attention to the lecture.

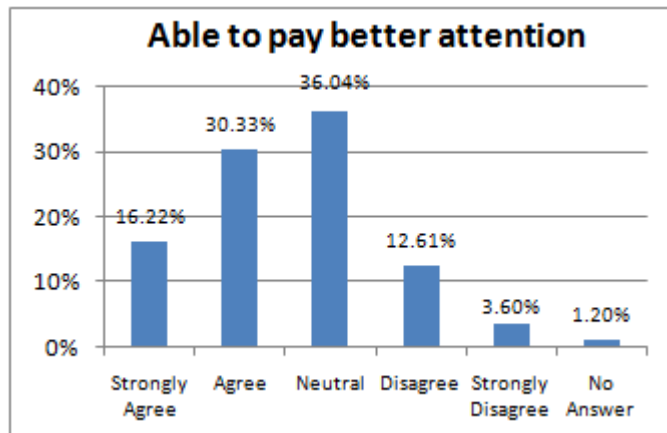


Figure A.4: The lecturer used the e-Fuzion technology effectively in class.

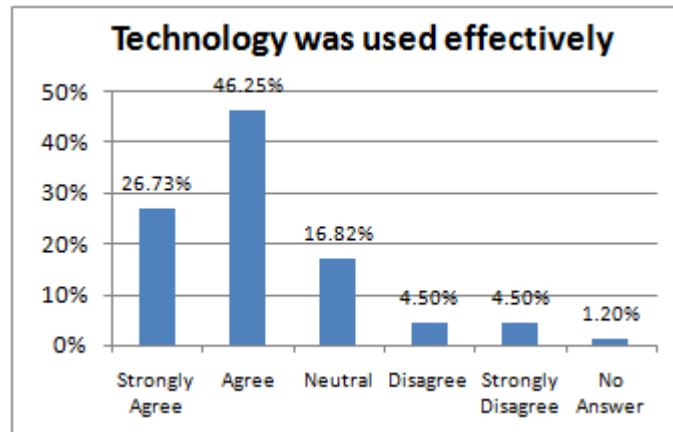


Figure A.5: This type of course is well-suited to e-Fuzion technology.

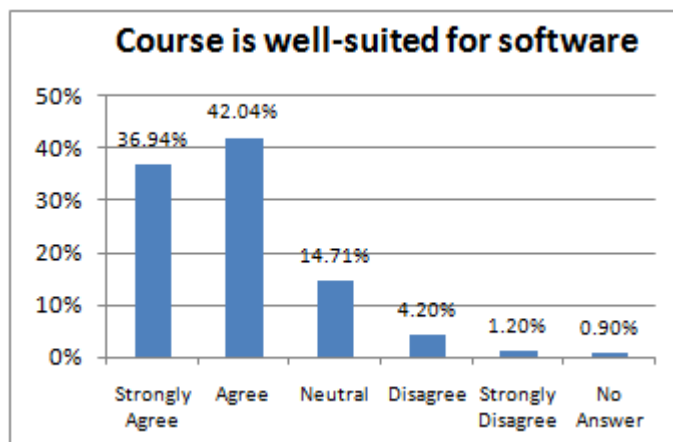


Figure A.6: Briefly describe your note-taking practices in classes not using e-Fuzion technology.

Previous note taking practice

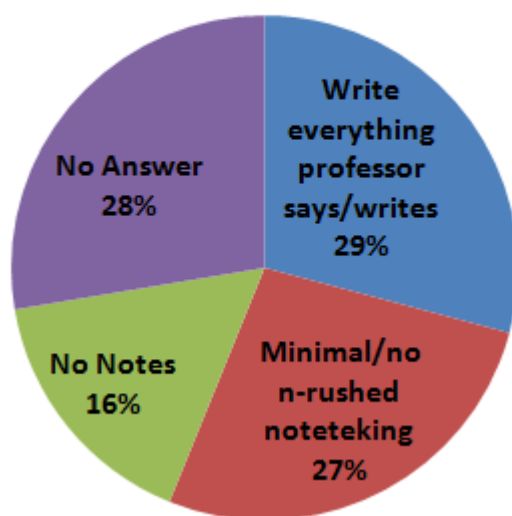


Figure A.7: Have your note-taking practices in this class changed as a result of e-Fuzion?

Have note-taking practices changed?

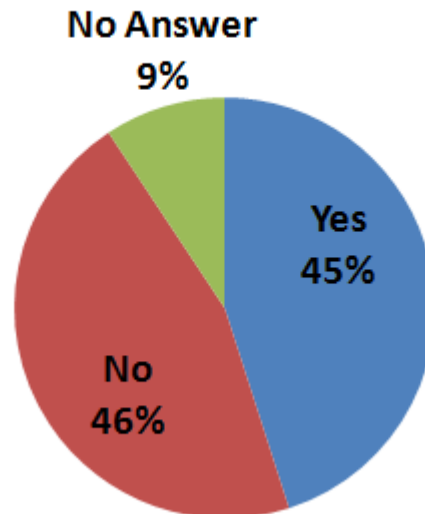


Figure A.8: Have you ever accessed the class Web page?

Ever accessed website?

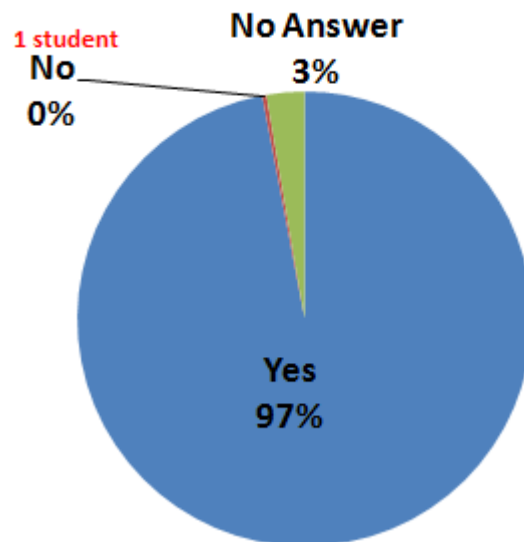


Figure A.9: How often did you access the captured lecture notes from class Web page?

How often notes were accessed

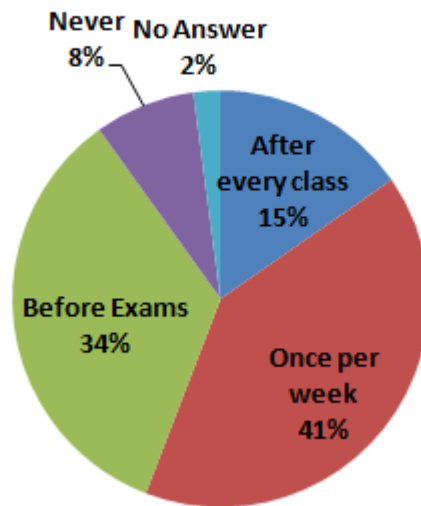


Figure A.10: Why did you access the captured lecture notes?

Reasons for accessing notes

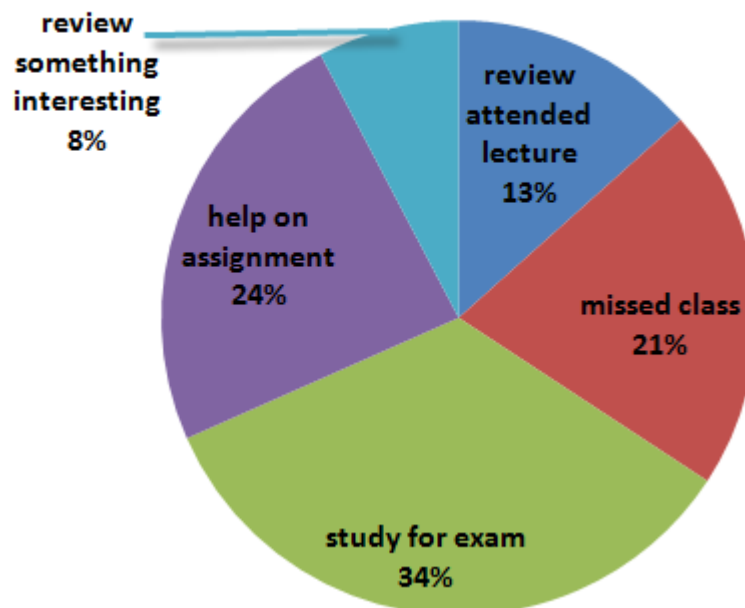


Figure A.11: All things being otherwise equal, I would prefer to take a class that uses e-Fuzion technology over the same class that does not.

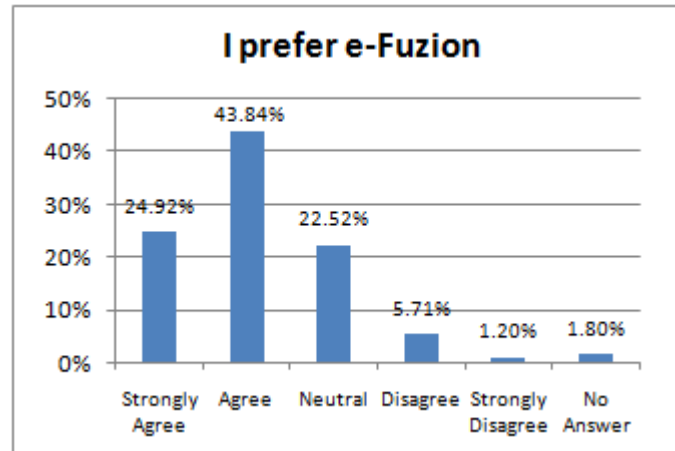


Figure A.12: e-Fuzion technology will encourage students to skip lectures.

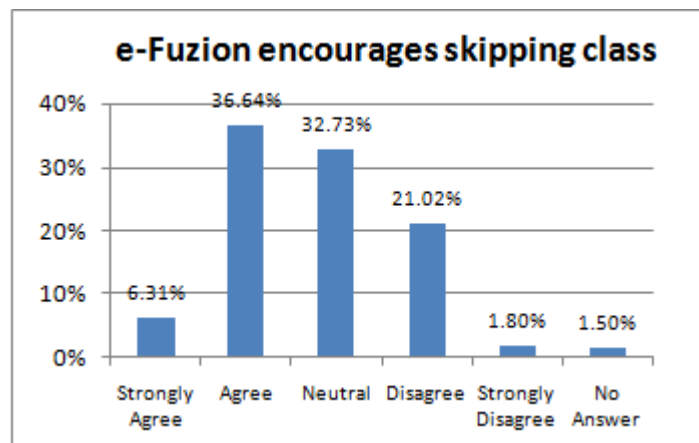


Figure A.13: e-Fuzion made me less worried about missing class.

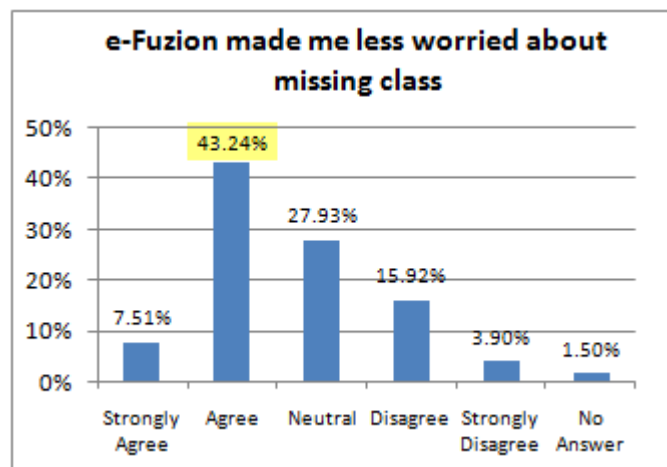


Figure A.14: I trust that captured lecture notes will be available after every class.

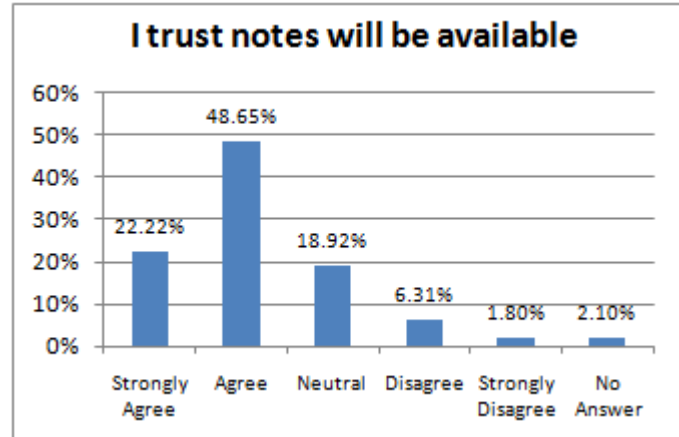


Figure A.15: Audio augmentation of the Web-based lecture notes would have increased their value to me.

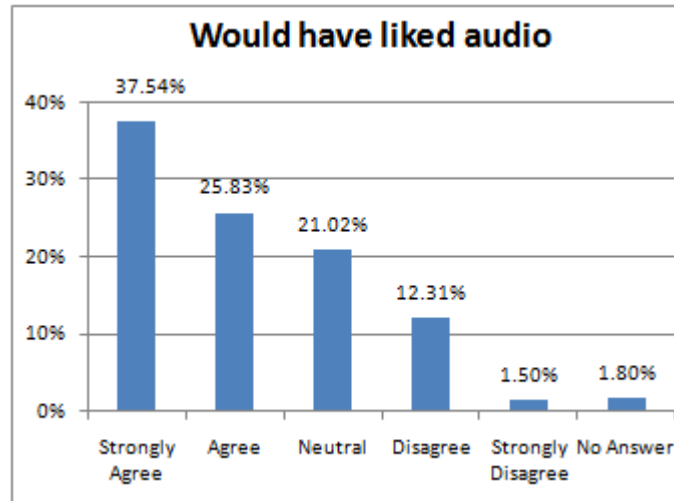


Figure A.16: Video augmentation of the Web-based lecture notes would have increased their value to me.

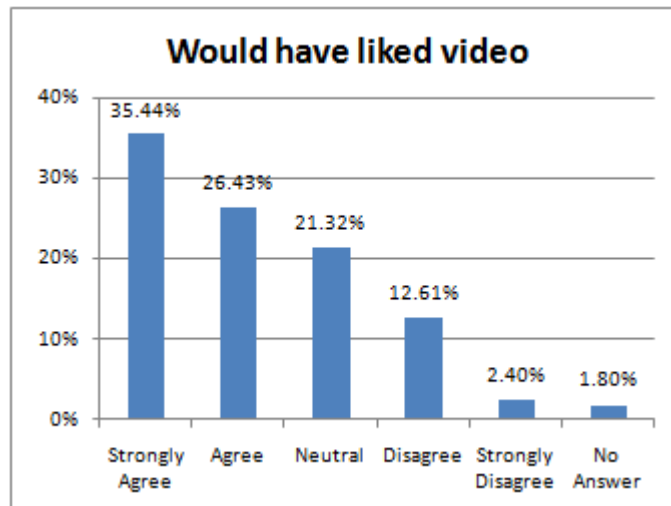


Figure A.17: Printing captured lecture slides after class for later review was valuable to me.

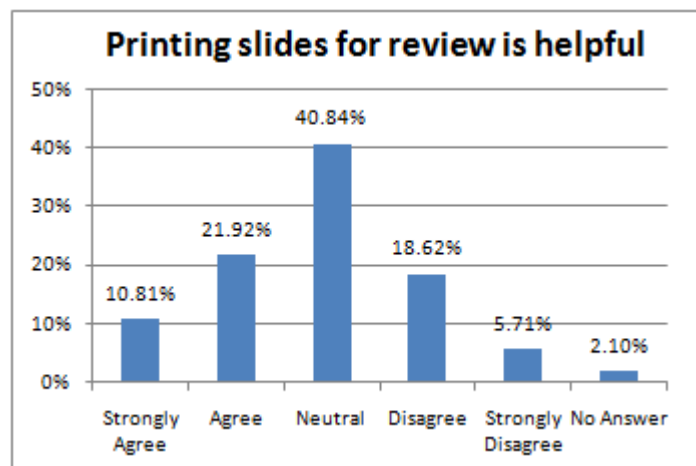


Figure A.18: The value of captured lecture notes would be greatly enhanced if replay of the instructor's ink was possible.

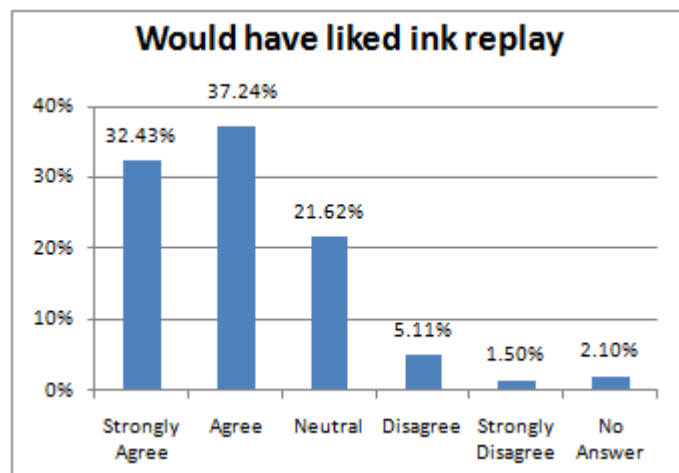


Figure A.19: What features of e-Fuzion did you find most useful?

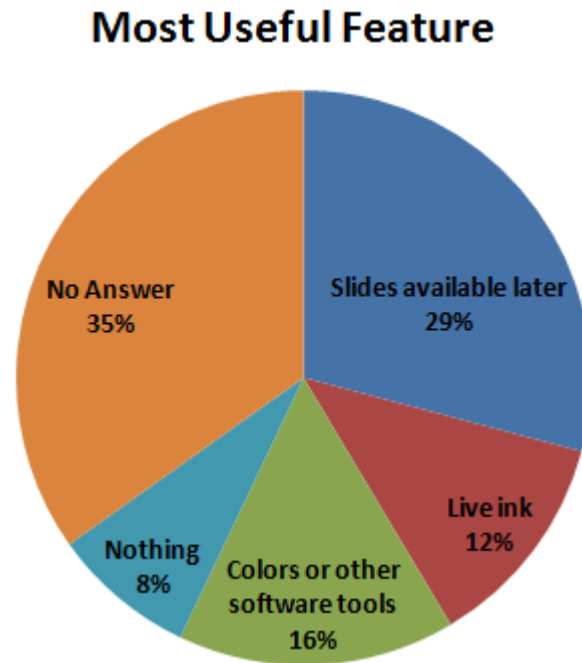


Figure A.20: What features inside the class did you find least useful, or even distracting?

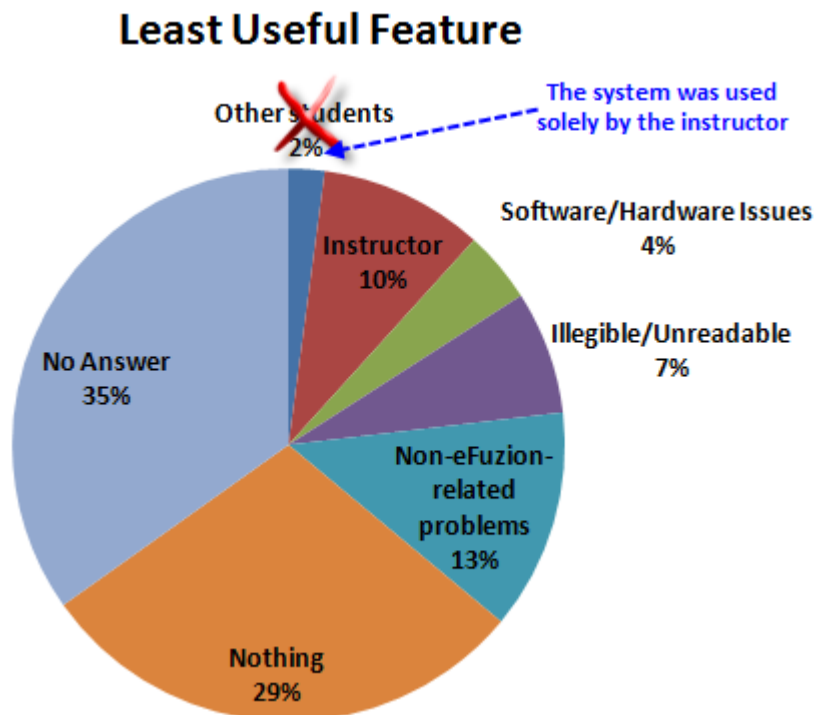


Figure A.21: What would you most like improved or added to capabilities of e-Fuzion?

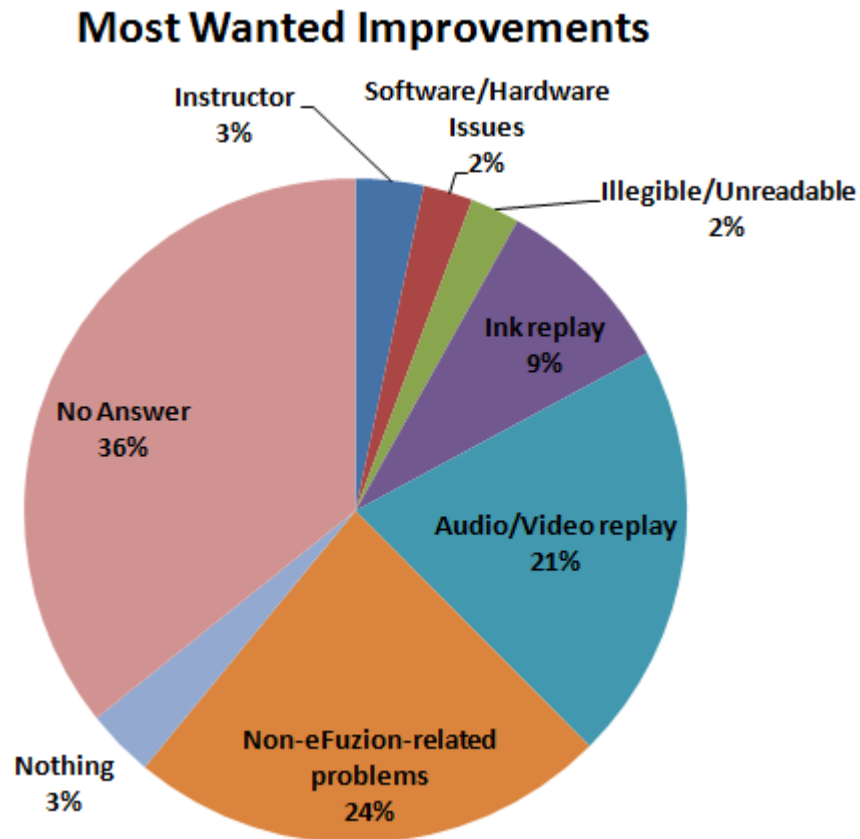


Figure A.22: If your class had prepared lectures, and you could print them before class and then take notes on top of them during the lecture, would/did you take notes this way?

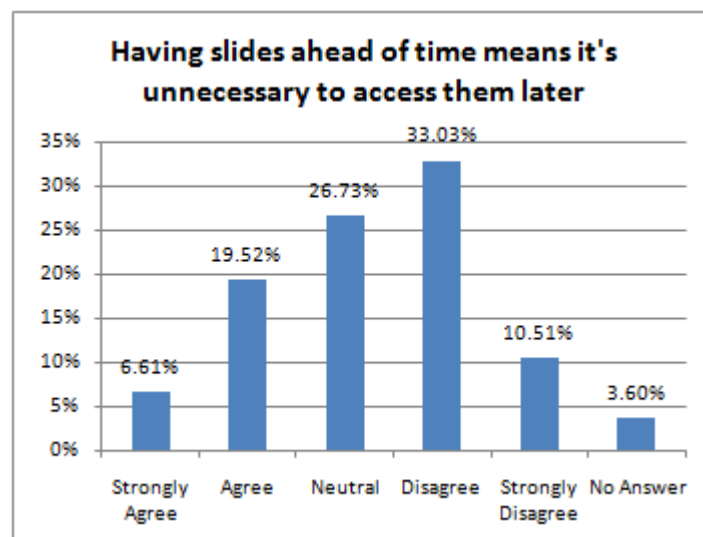


Figure A.23: The availability of prepared lectures (printed out) before class would make it unnecessary to access the captured notes after class.

Would you print slides before class?

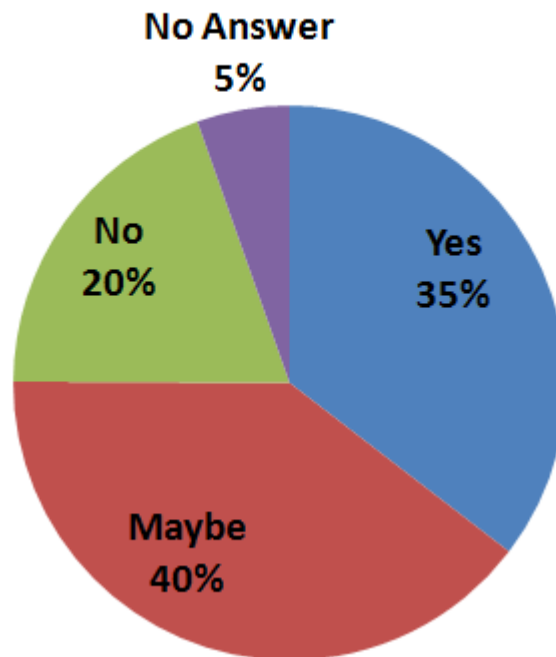
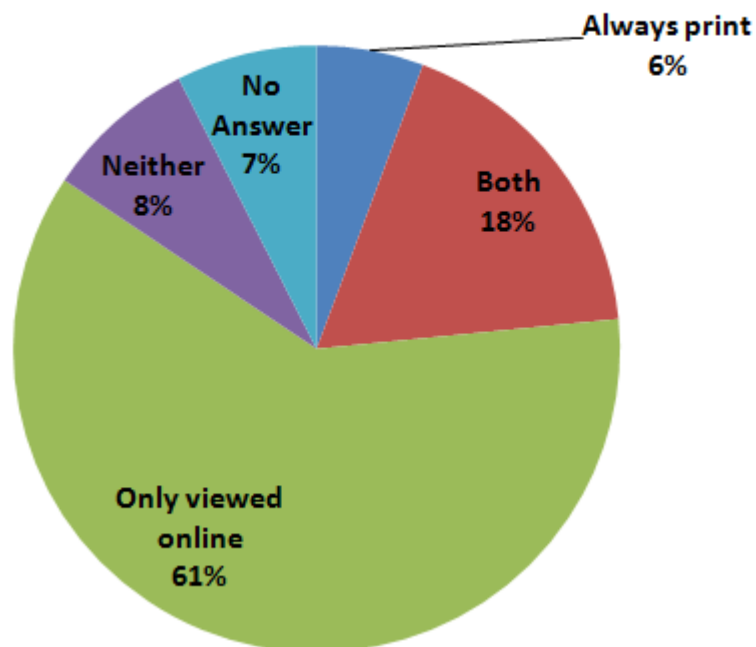


Figure A.24: If you accessed the captured notes, were you more likely to print them out instead of view them online?

Print notes vs. view online



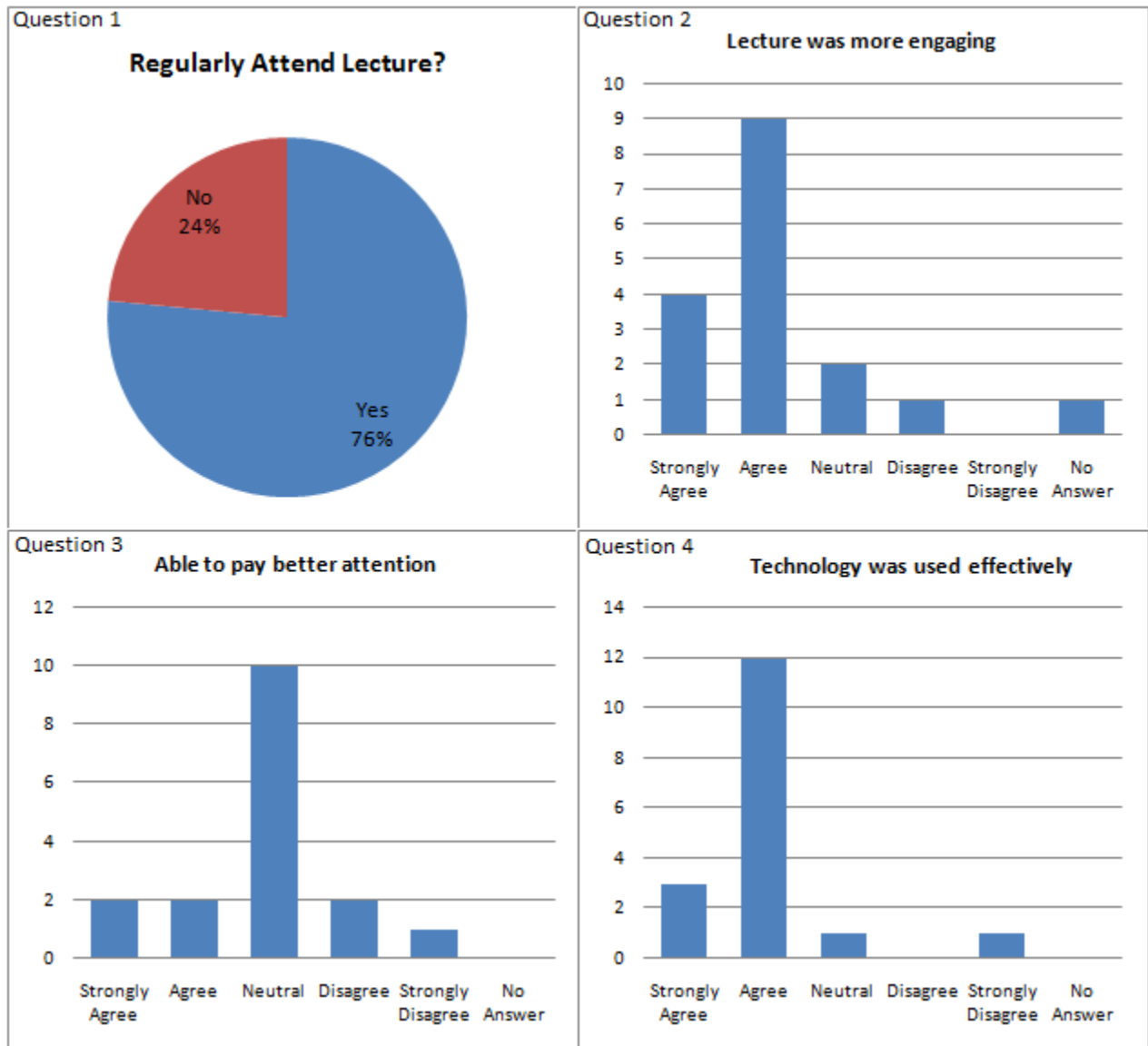
A.2 Chapter 6 Survey Questions and Results

Q1 Did you regularly attend lecture?

Q2 The classroom lecture was more engaging or interesting to me as result of the use of technology in class and the availability of notes afterwards.

Q3 Because captured lecture notes are available after class, I was able to better pay attention to the lecture.

Q4 The lecturer used the e-Fuzion technology effectively in class.

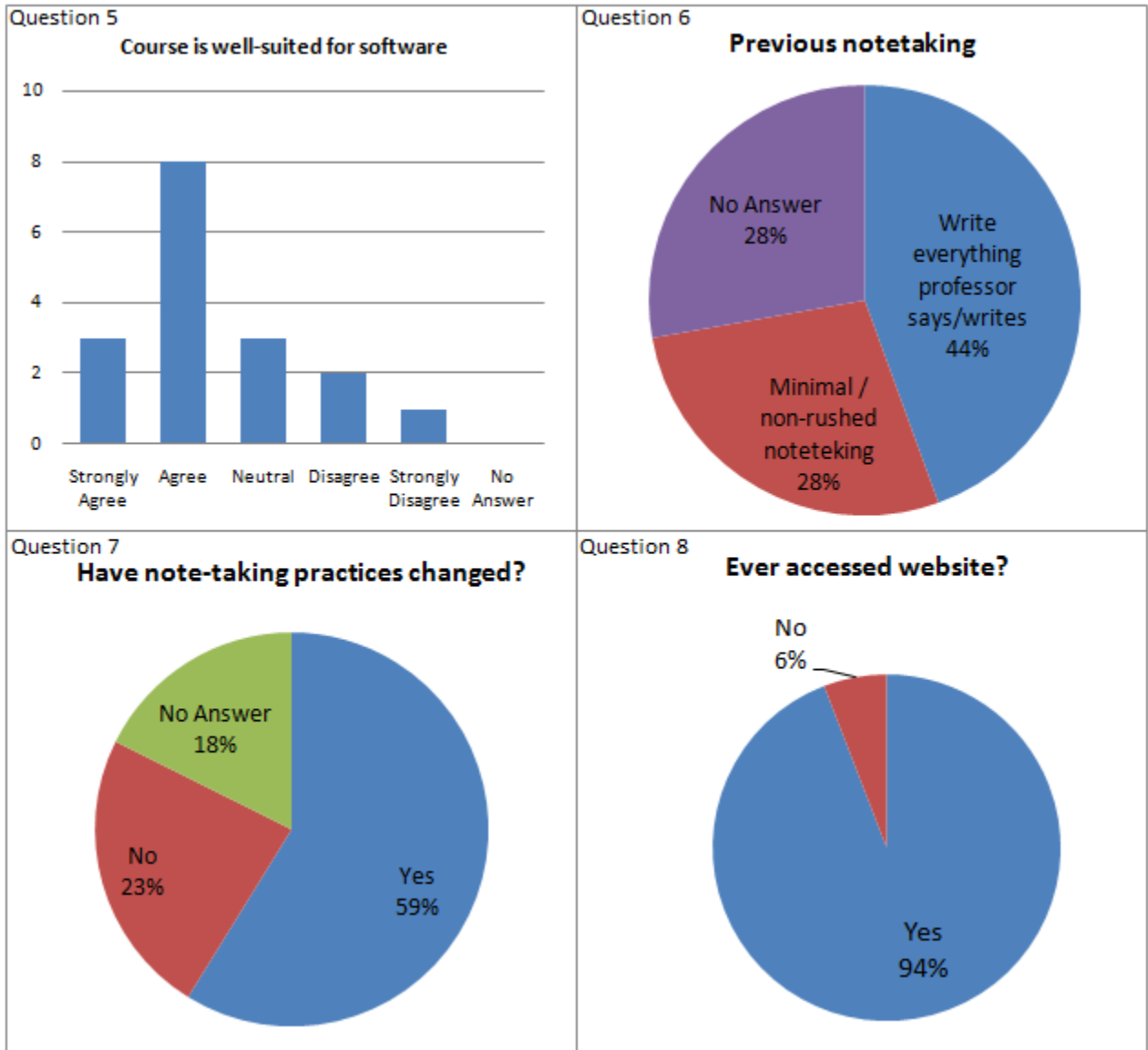


Q5 This type of course is well-suited to e-Fuzion technology.

Q6 Briefly describe your note-taking practices in classes not using e-Fuzion technology.

Q7 Have your note-taking practices in this class changed as a result of e-Fuzion?

Q8 Have you ever accessed the class Web page?

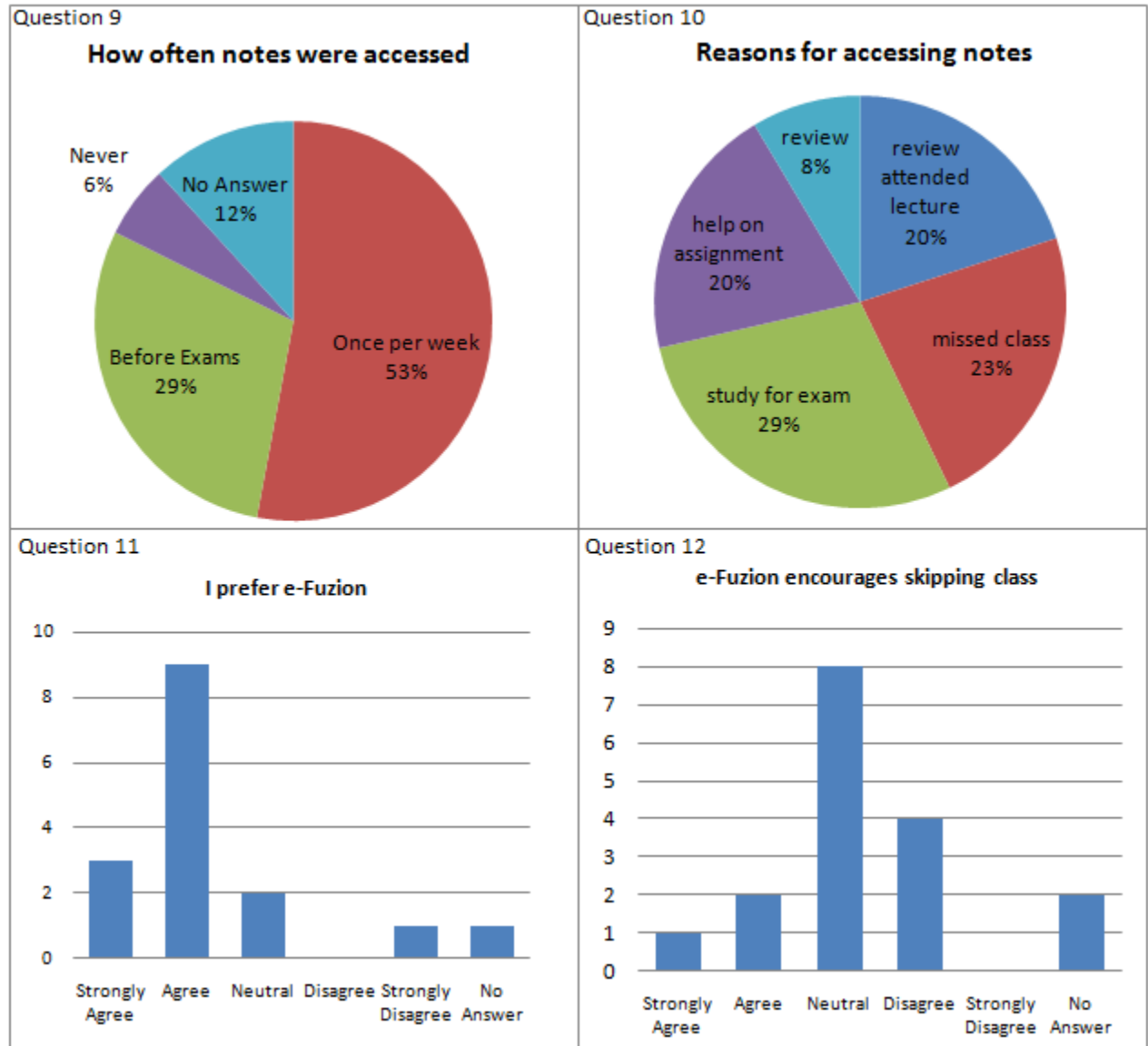


Q9 How often did you access the captured lecture notes from class Web page?

Q10 The lecturer used the e-Fuzion technology effectively in class.

Q11 All things being otherwise equal, I would prefer to take a class that uses e-Fuzion technology over the same class that does not.

Q12 e-Fuzion technology will encourage students to skip lectures.

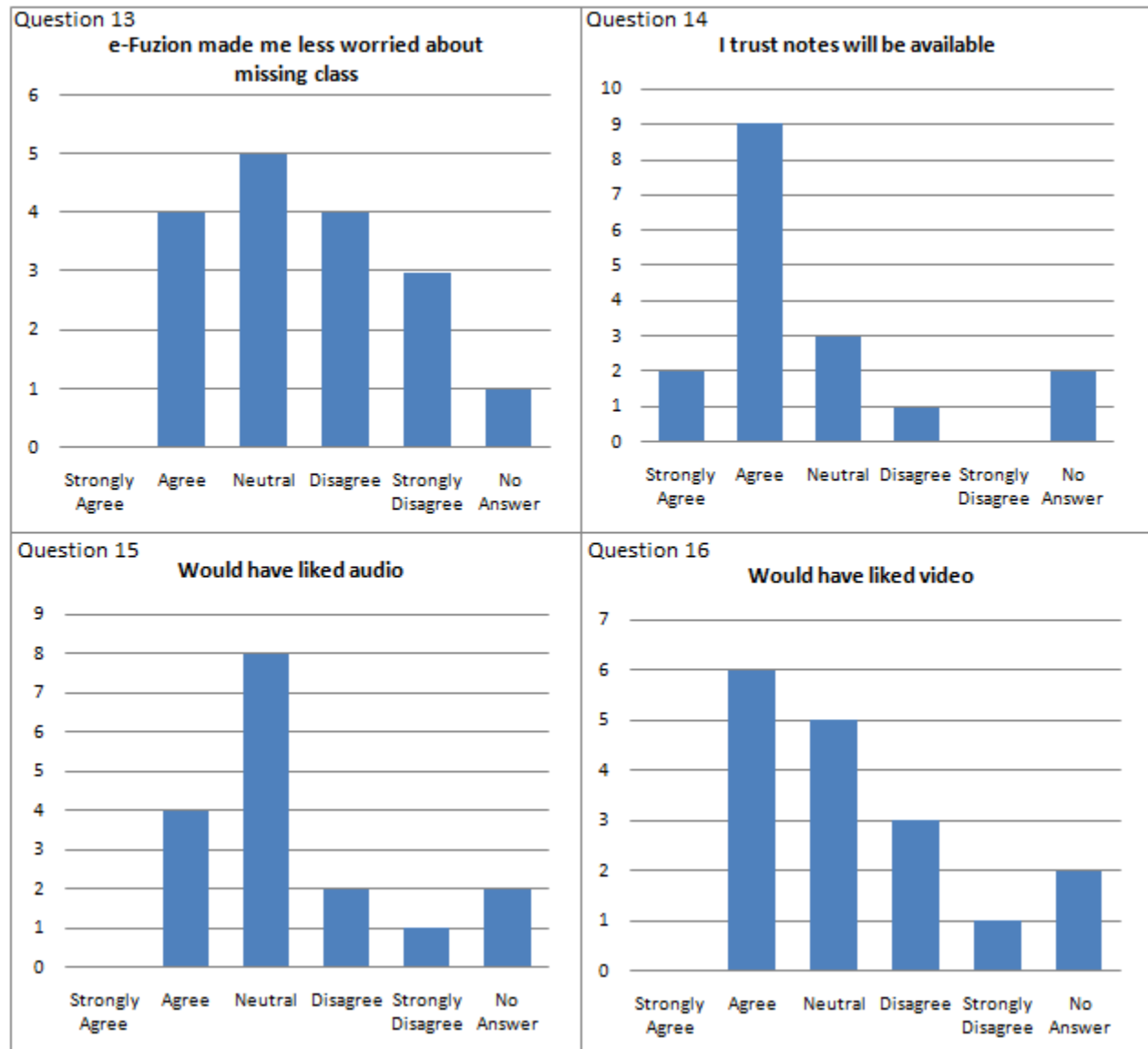


Q13 e-Fuzion made me less worried about missing class.

Q14 I trust that captured lecture notes will be available after every class.

Q15 Audio augmentation of the Web-based lecture notes would have increased their value to me.

Q16 Video augmentation of the Web-based lecture notes would have increased their value to me.

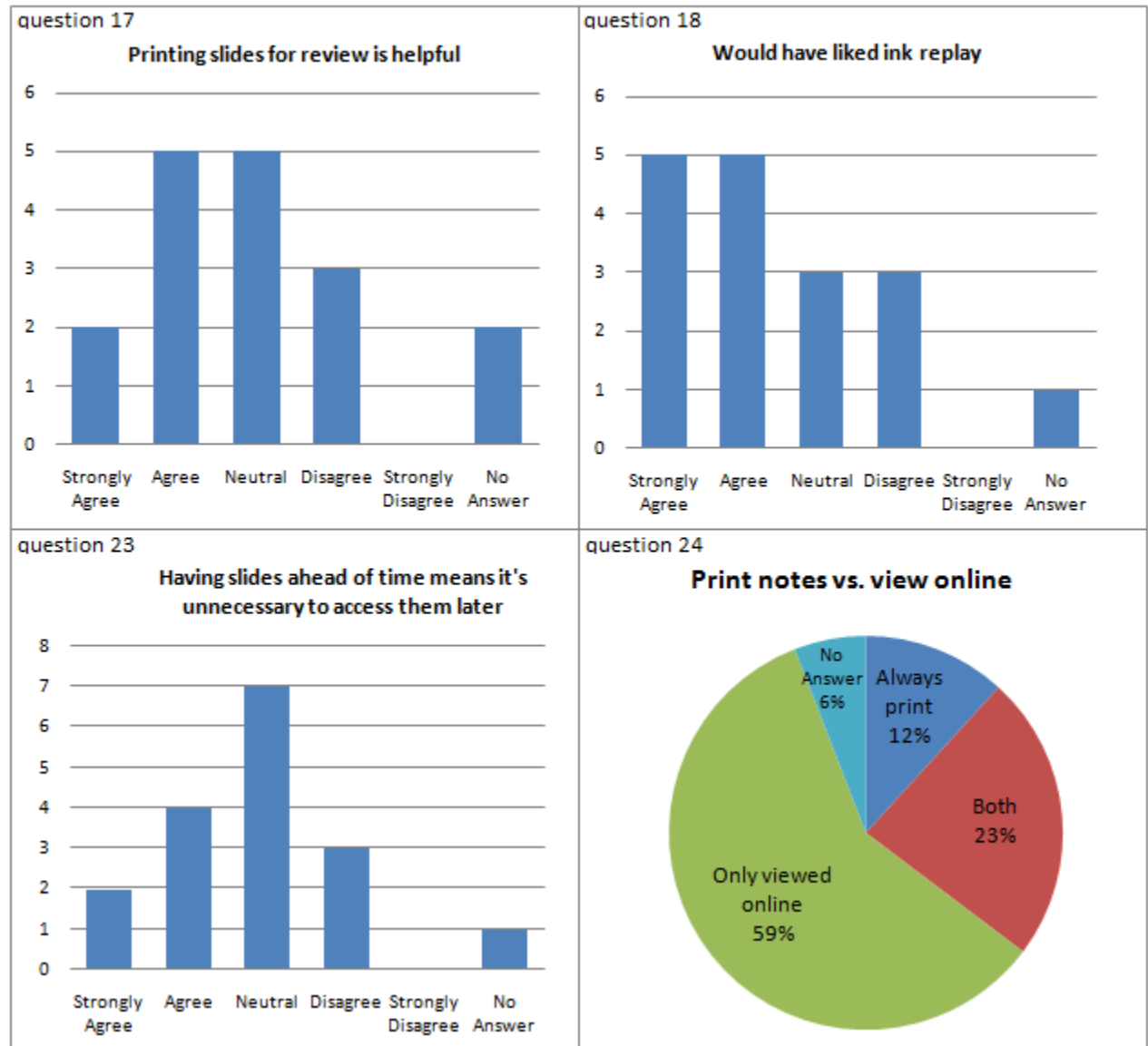


Q17 Printing captured lecture slides after class for later review was valuable to me.

Q18 The value of captured lecture notes would be greatly enhanced if replay of the instructor's ink was possible.

Q23 The availability of prepared lectures (printed out) before class would make it unnecessary to access the captured notes after class.

Q24 If you accessed the captured notes, were you more likely to print them out instead of view them online?



A.3 Chapter 7 Survey Questions and Results

1. Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them? cs125tpc A.25 alg1tech A.42 alg1noTech A.76 algebra2 A.59 geometryNoTech A.88 geometryTech A.100
2. In those cases when you have a question but don't ask it out loud, how often does another student ask it? cs125tpc A.26 alg1tech A.43 alg1noTech A.77 algebra2 A.60 geometryNoTech A.89 geometryTech A.101
3. When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking? cs125tpc A.27 alg1tech A.44 alg1noTech A.78 algebra2 A.61 geometryNoTech A.90 geometryTech A.102
4. How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)? cs125tpc A.28 alg1tech A.45 alg1noTech A.79 algebra2 A.62 geometryNoTech A.91 geometryTech A.103
5. How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)? cs125tpc A.29 alg1tech A.46 alg1noTech A.80 algebra2 A.63 geometryNoTech A.92 geometryTech A.104
6. How often do you feel the instructor offers you one on one attention during class? cs125tpc A.30 alg1tech A.47 alg1noTech A.81 algebra2 A.64 geometryNoTech A.93 geometryTech A.105
7. How often do you feel the instructor is able to identify that you are struggling with a particular topic? cs125tpc A.31 alg1tech A.48 alg1noTech A.82 algebra2 A.65 geometryNoTech A.94 geometryTech A.106
8. How often do you feel the instructor is able to identify that you understand the material? cs125tpc A.32 alg1Tech A.49 alg1noTech A.83 algebra2 A.66 geometryNoTech A.95 geometry A.95
9. How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material? cs125tpc A.33 alg1Tech A.50 alg1noTech A.84 algebra2 A.67 geometryNoTech A.96 geometryTech A.108
10. Small group exercises are effective for improving my understanding of material. cs125tpc A.34 alg1Tech A.51 alg1noTech A.85 algebra2 A.68 geometryNoTech A.97 geometryTech A.109
11. The in-class activities keep me interested in the course content. cs125tpc A.35 alg1Tech alg1noTech A.86 algebra2 A.69 geometryNoTech A.98 geometryTech A.110
12. My small group needs more instructor guidance to work effectively. cs125tpc A.36 alg1Tech alg1noTech A.87 algebra2 A.70 geometryNoTech A.99 geometryTech A.111
13. There is a positive interaction between students and instructor. cs125tpc A.37 alg1Tech A.54 algebra2 A.71
14. The instructor promotes an atmosphere conducive to work and learning. cs125tpc A.38 alg1Tech A.55 algebra2 A.72
15. The instructor promotes interaction amongst the students. cs125tpc A.39 alg1Tech A.56 algebra2 A.73
16. The instructor encourages class members to work as a team. cs125tpc A.40 alg1Tech A.57 algebra2 A.74
17. How often do you seek out the teacher to ask a question outside of class? cs125tpc A.41 alg1Tech A.58 algebra2 A.75

CS125TPC

Figure A.25: Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them?

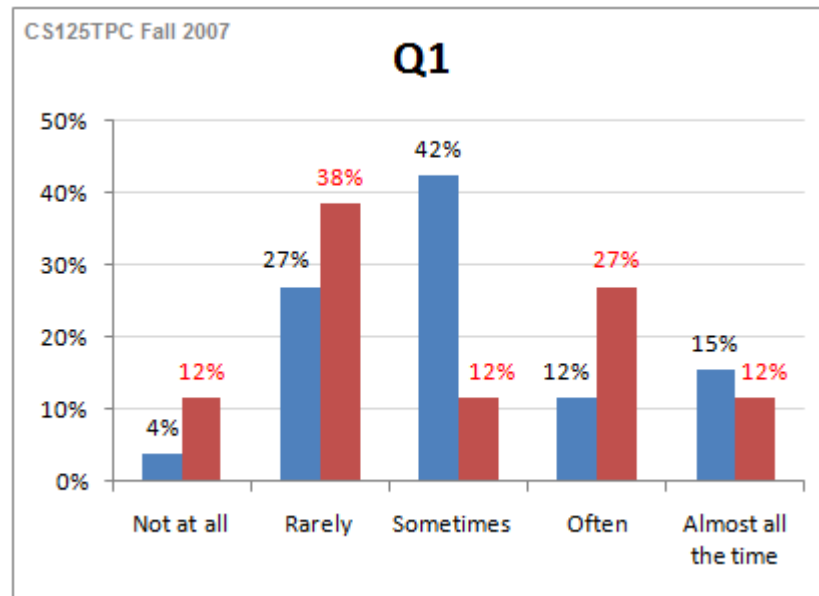


Figure A.26: In those cases when you have a question but don't ask it out loud, how often does another student ask it?

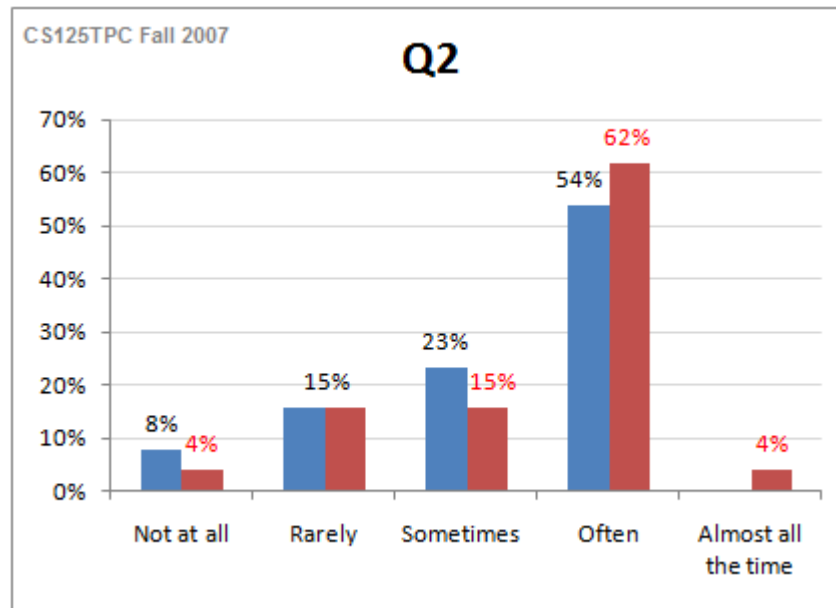


Figure A.27: When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking?

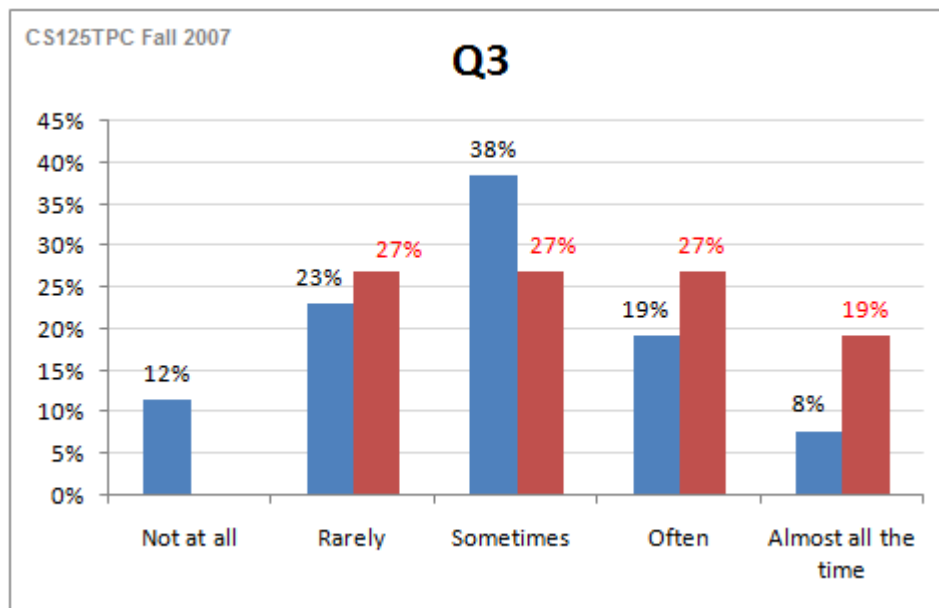


Figure A.28: How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)?

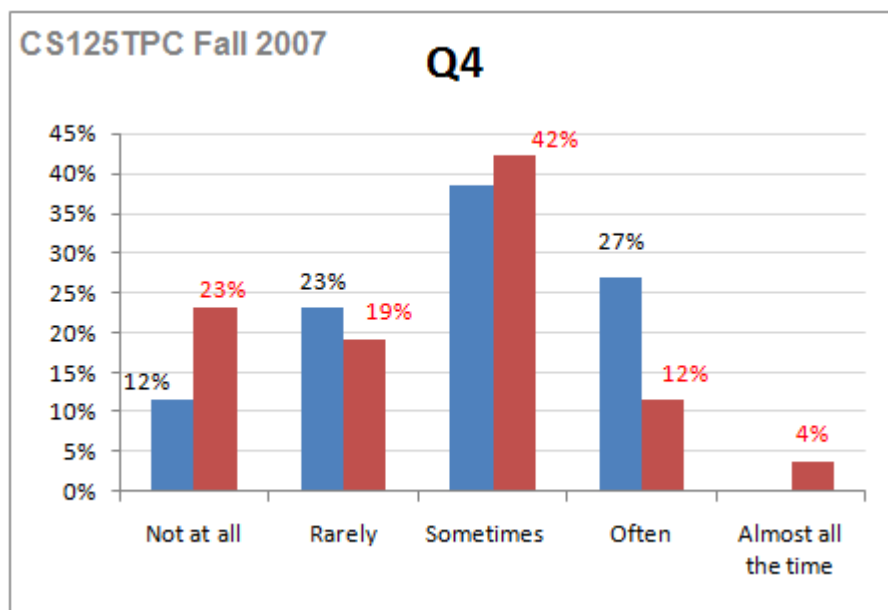


Figure A.29: How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)?

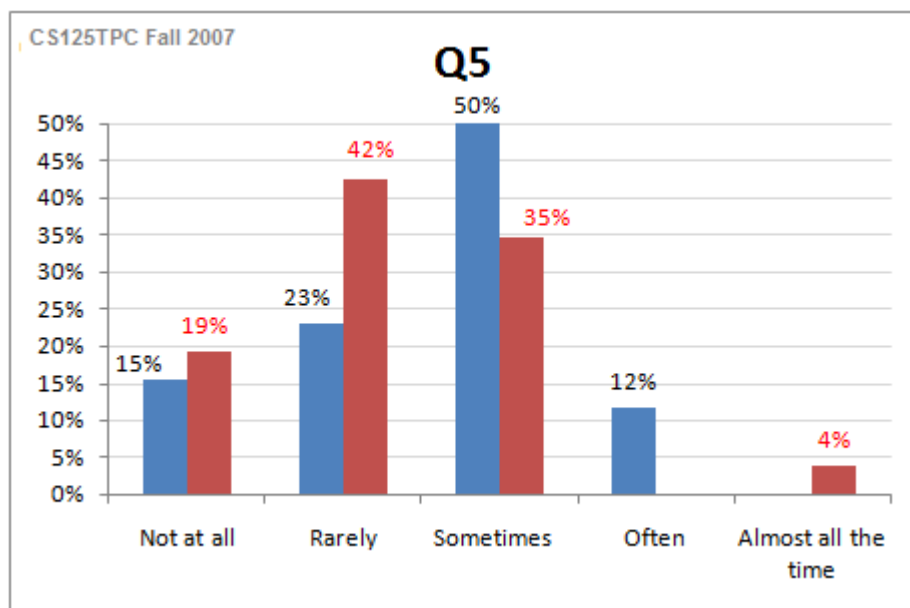


Figure A.30: How often do you feel the instructor offers you one on one attention during class?

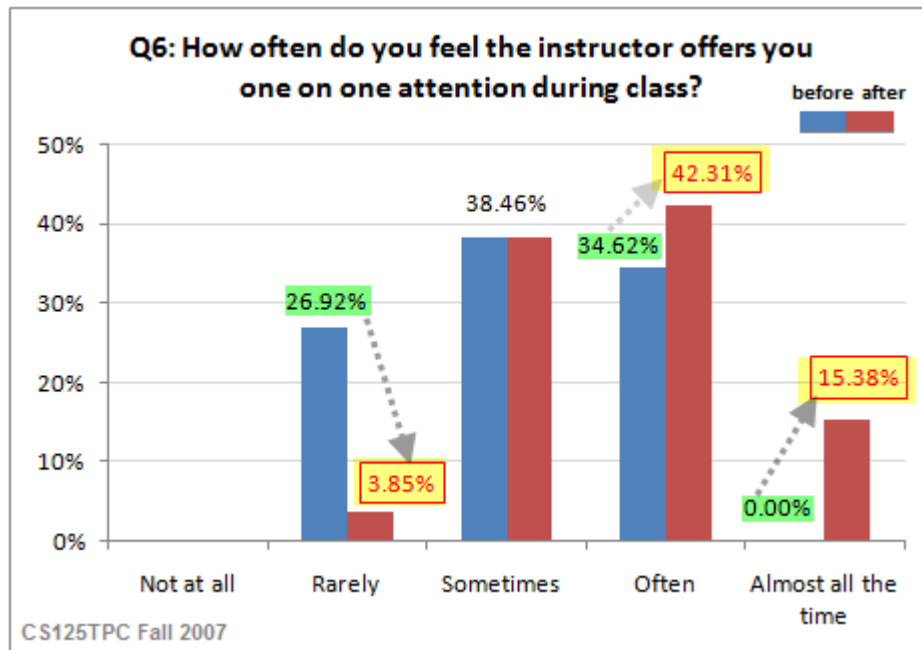


Figure A.31: How often do you feel the instructor is able to identify that you are struggling with a particular topic?

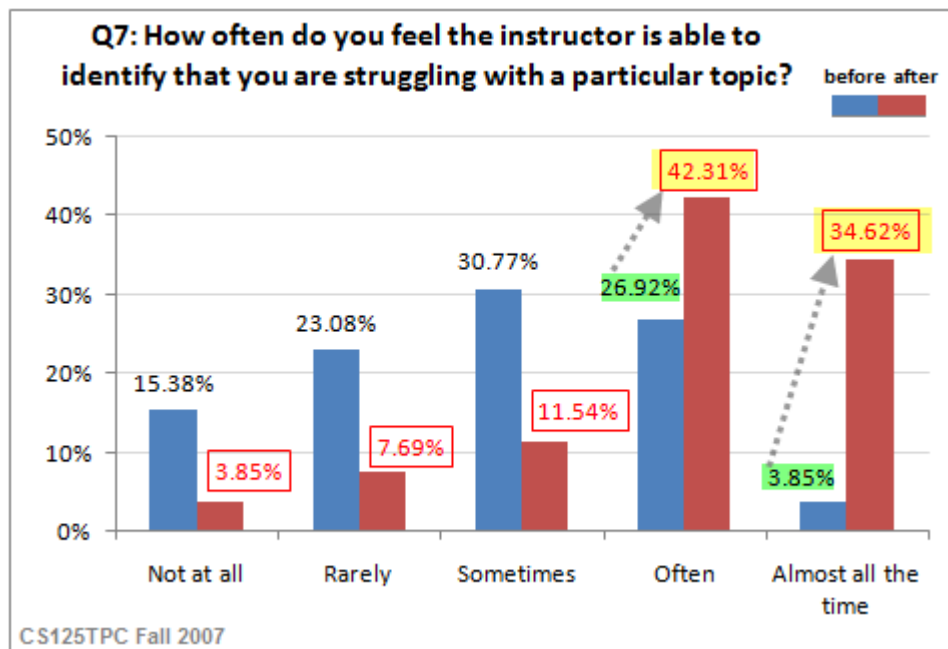


Figure A.32: How often do you feel the instructor is able to identify that you understand the material?

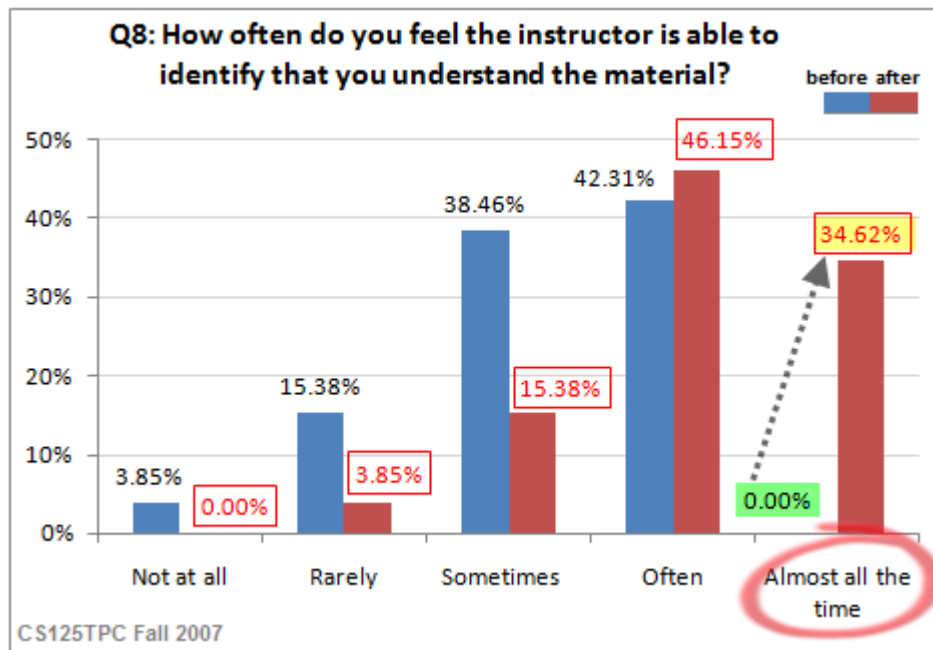


Figure A.33: How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

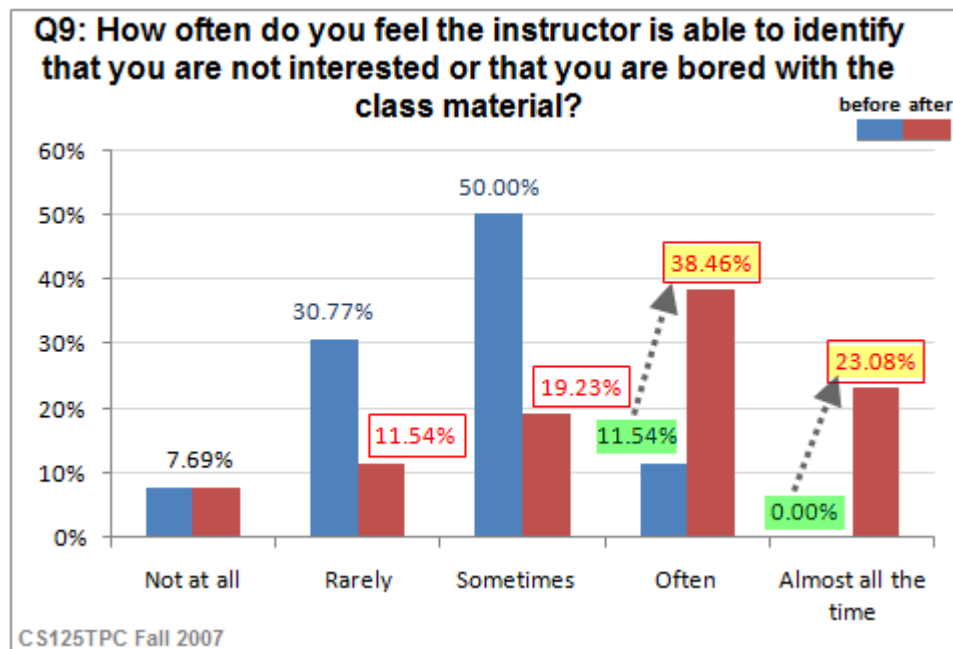


Figure A.34: Small group exercises are effective for improving my understanding of material.

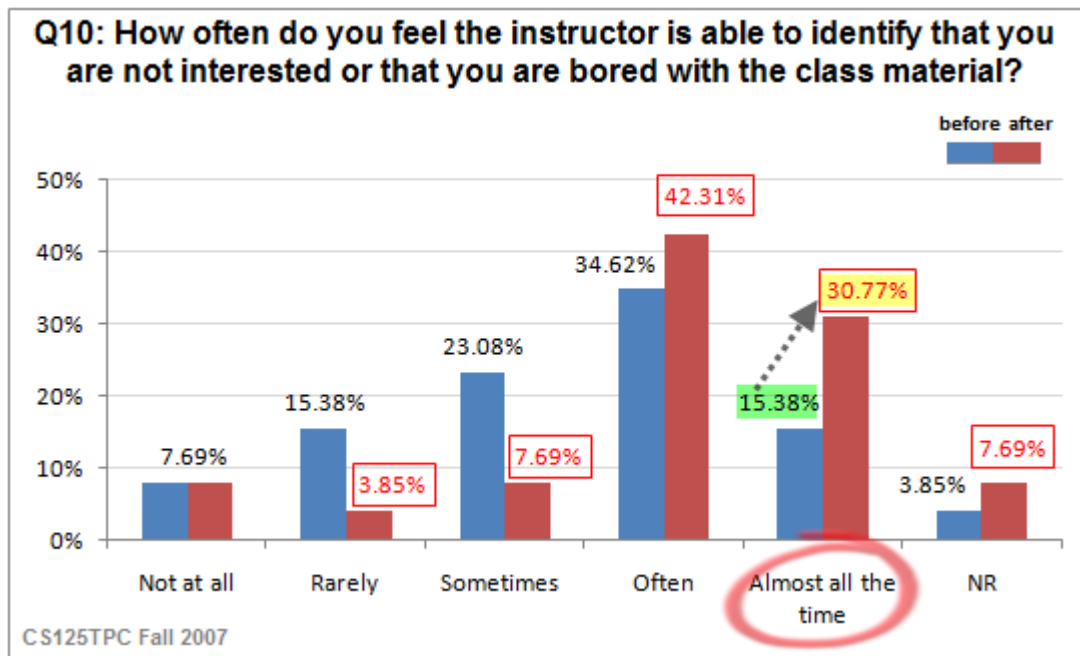


Figure A.35: The in-class activities keep me interested in the course content.

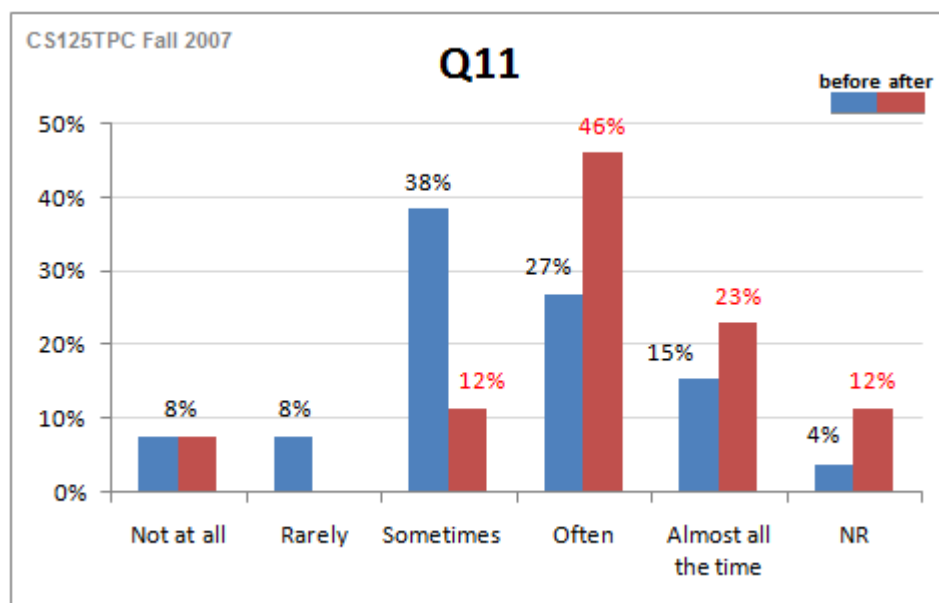


Figure A.36: My small group needs more instructor guidance to work effectively.

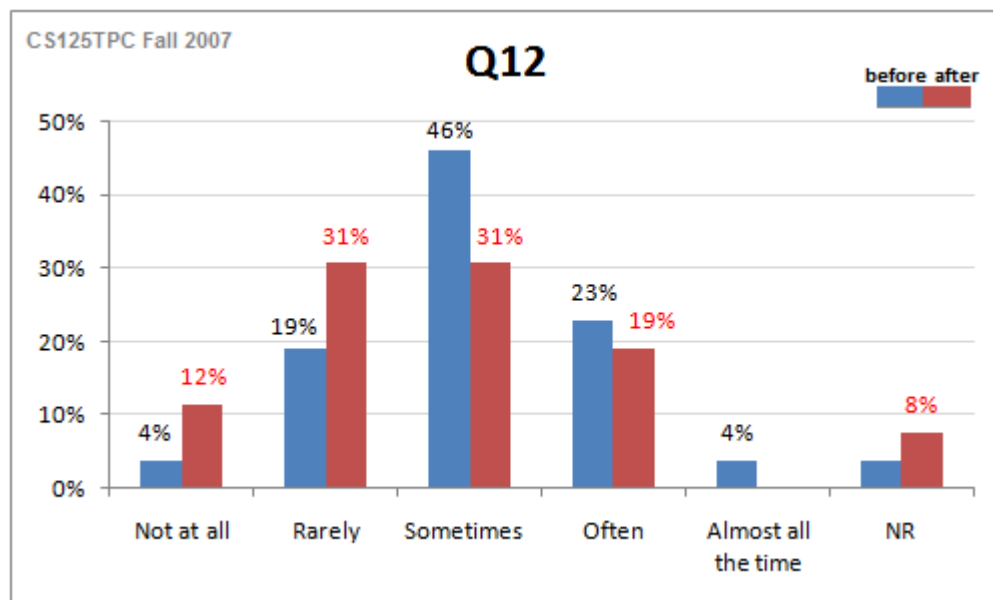
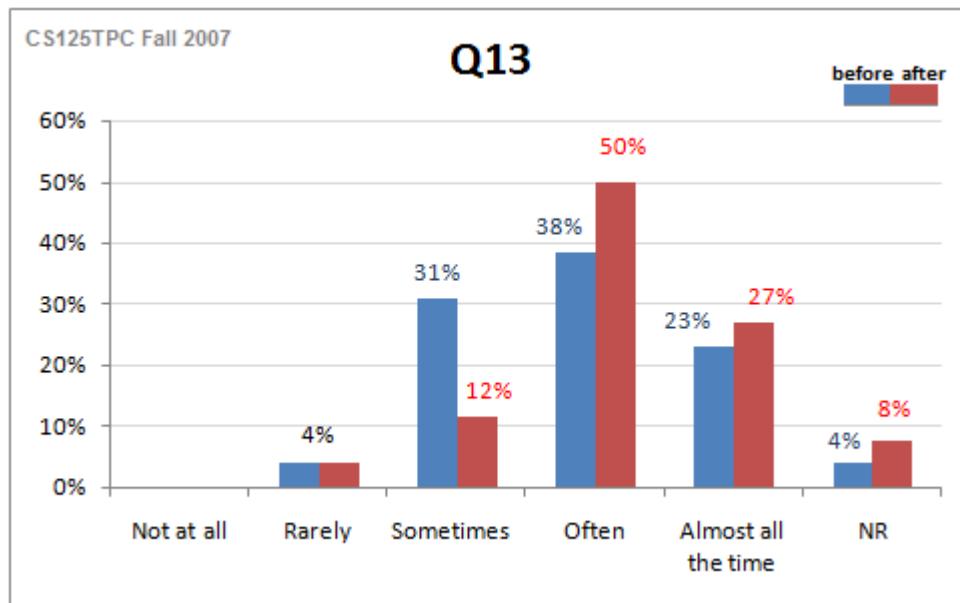


Figure A.37: There is a positive interaction between students and instructor.



Algebra 1

Figure A.38: The instructor promotes an atmosphere conducive to work and learning.

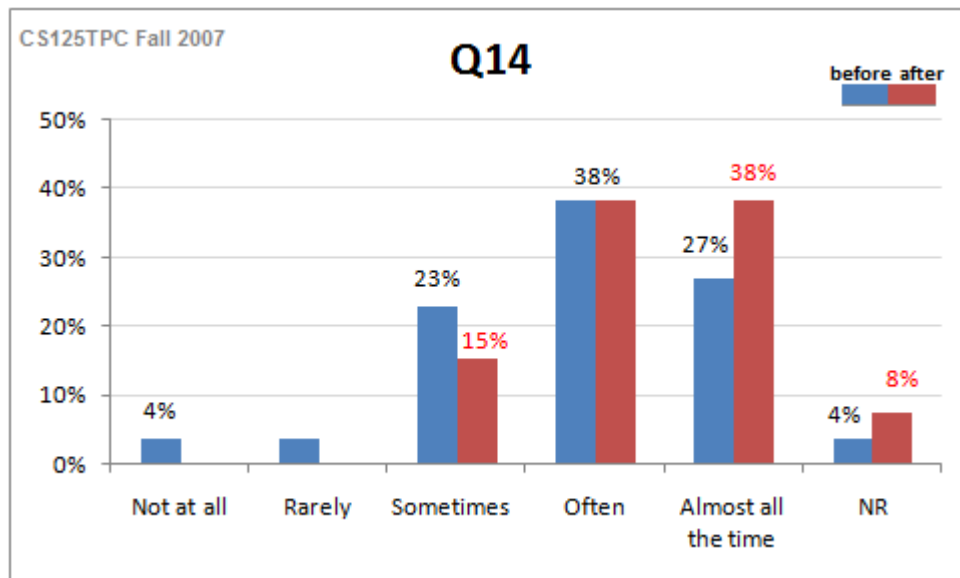


Figure A.39: The instructor promotes interaction amongst the students.

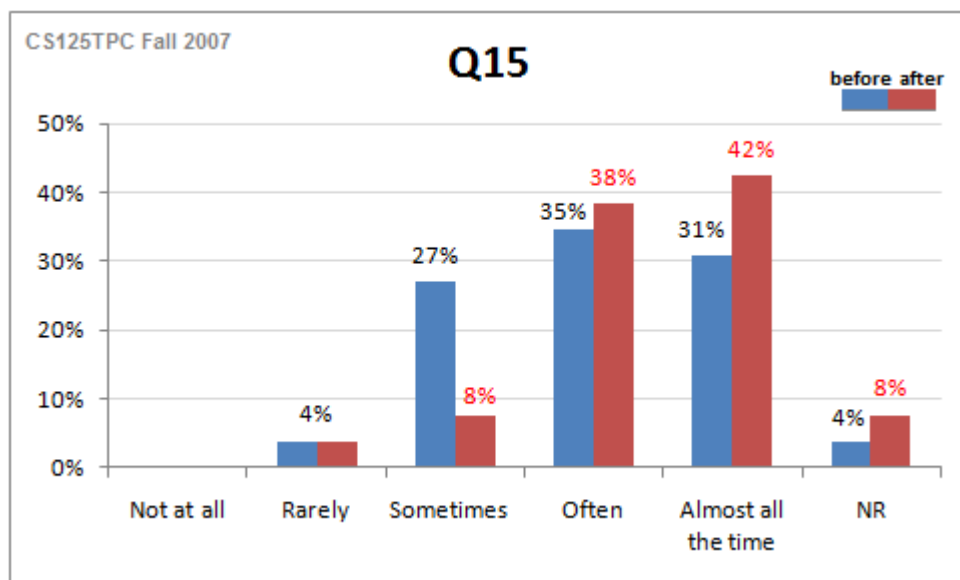


Figure A.40: The instructor encourages class members to work as a team.

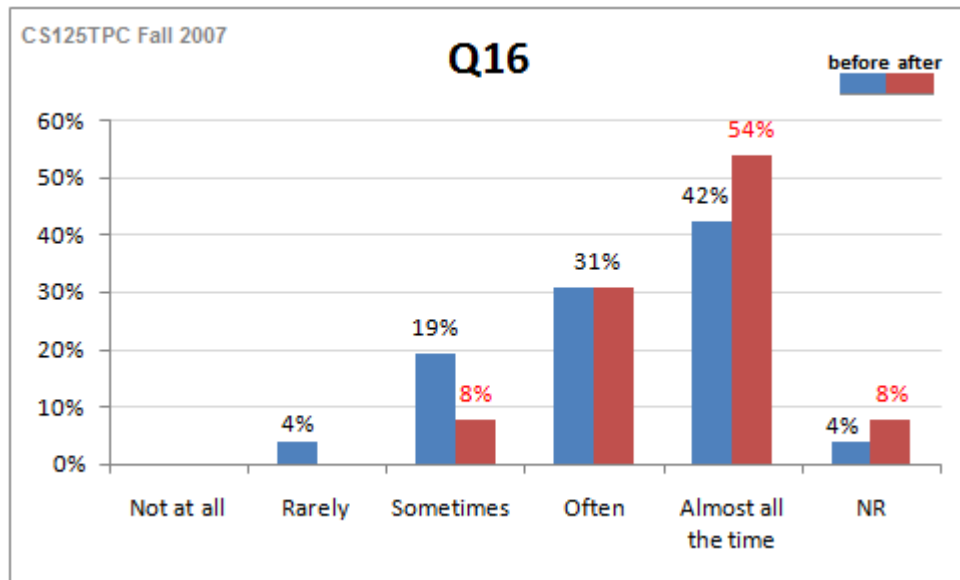


Figure A.41: How often do you seek out the teacher to ask a question outside of class?

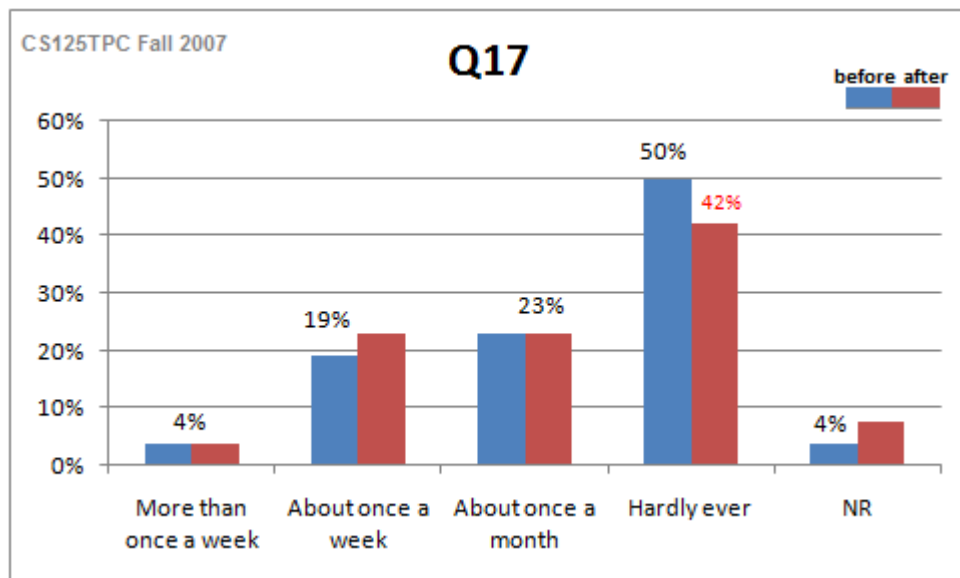


Figure A.42: Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them? ?

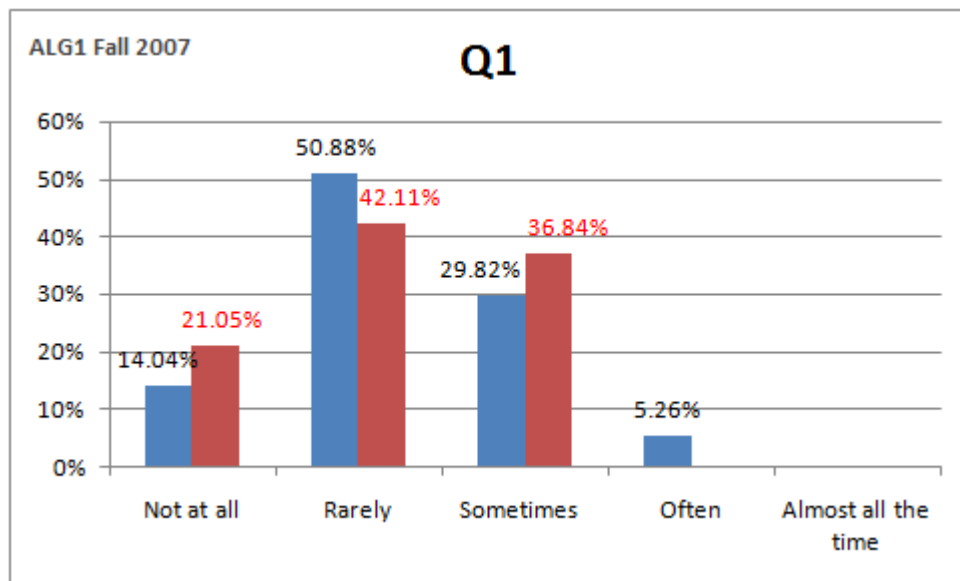


Figure A.43: In those cases when you have a question but don't ask it out loud, how often does another student ask it?

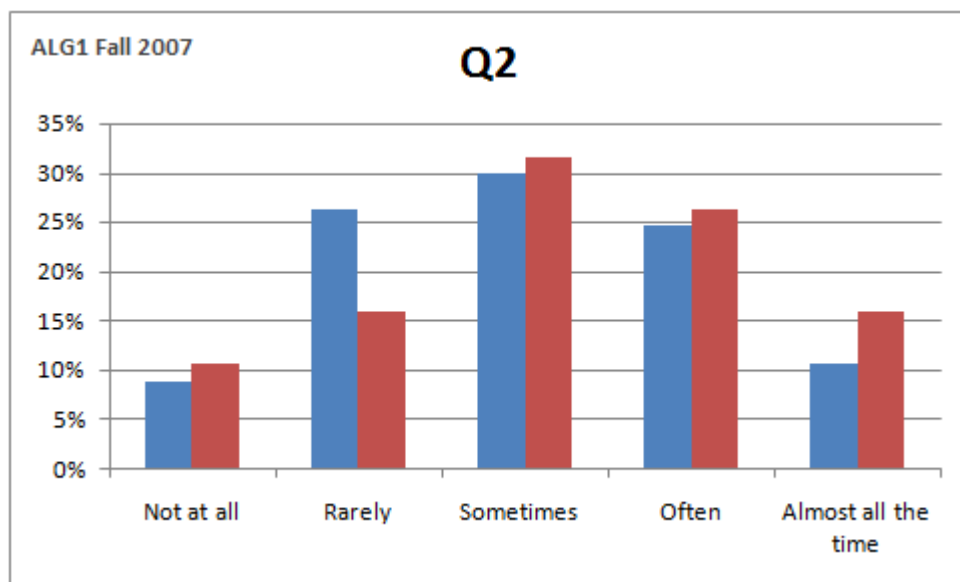


Figure A.44: When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking?

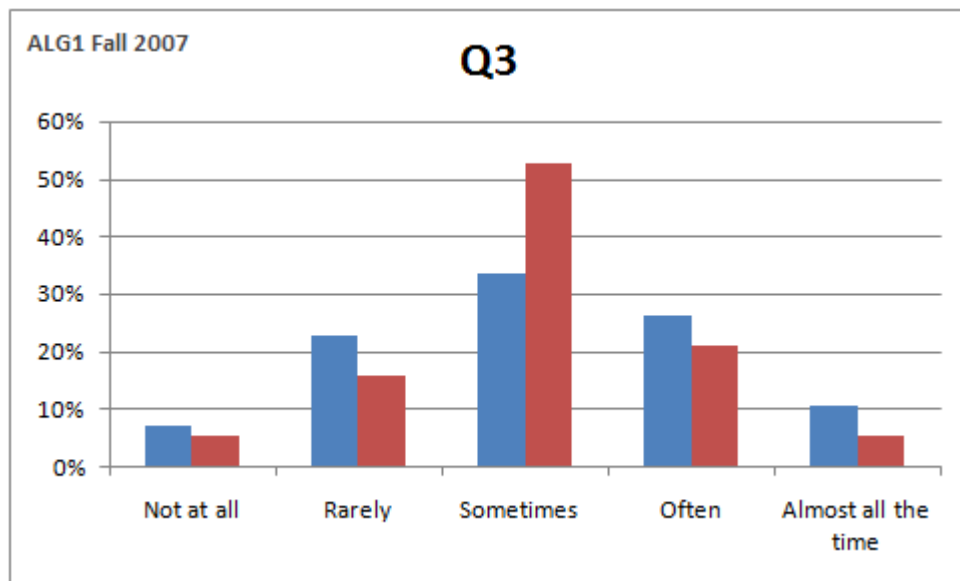


Figure A.45: How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)?

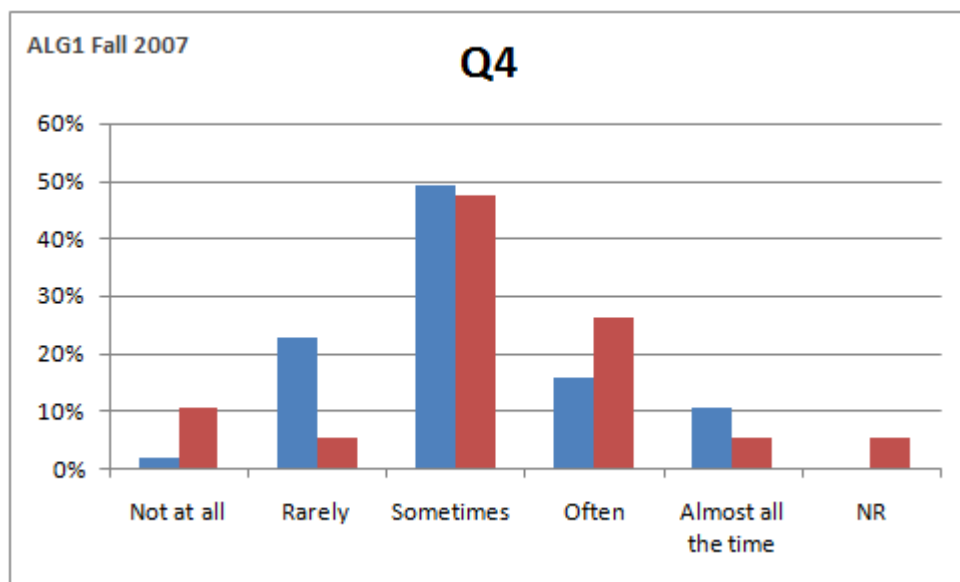


Figure A.46: How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)?

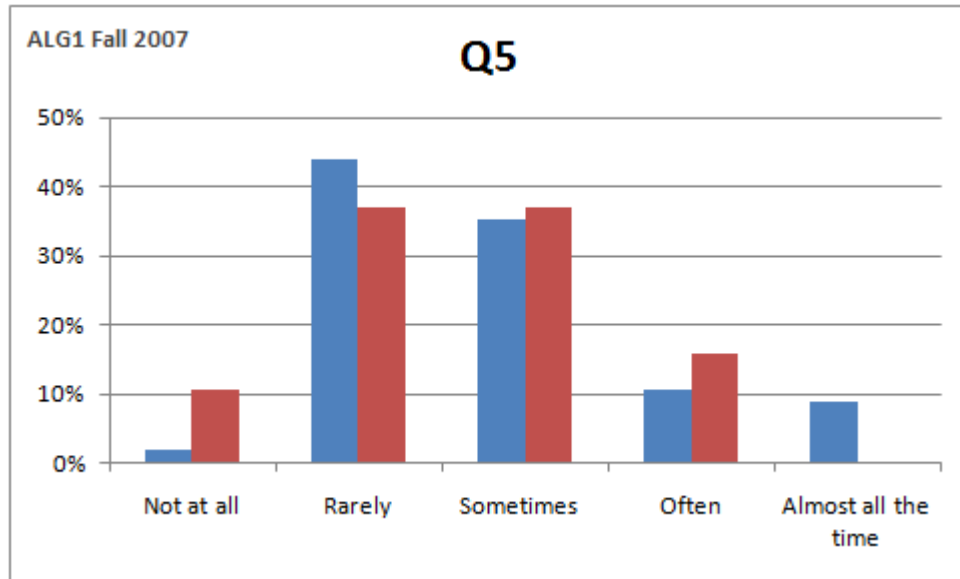


Figure A.47: How often do you feel the instructor offers you one on one attention during class?

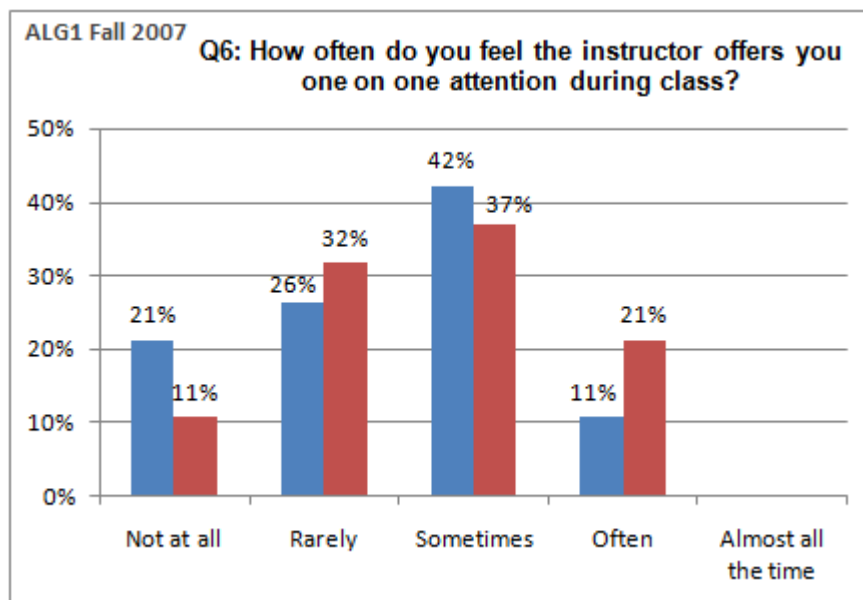


Figure A.48: How often do you feel the instructor is able to identify that you are struggling with a particular topic?

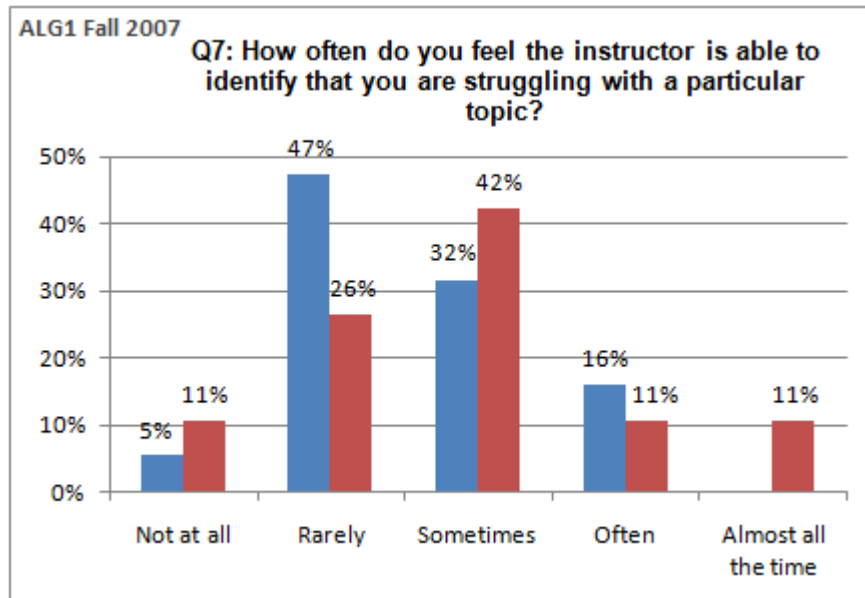


Figure A.49: How often do you feel the instructor is able to identify that you understand the material?

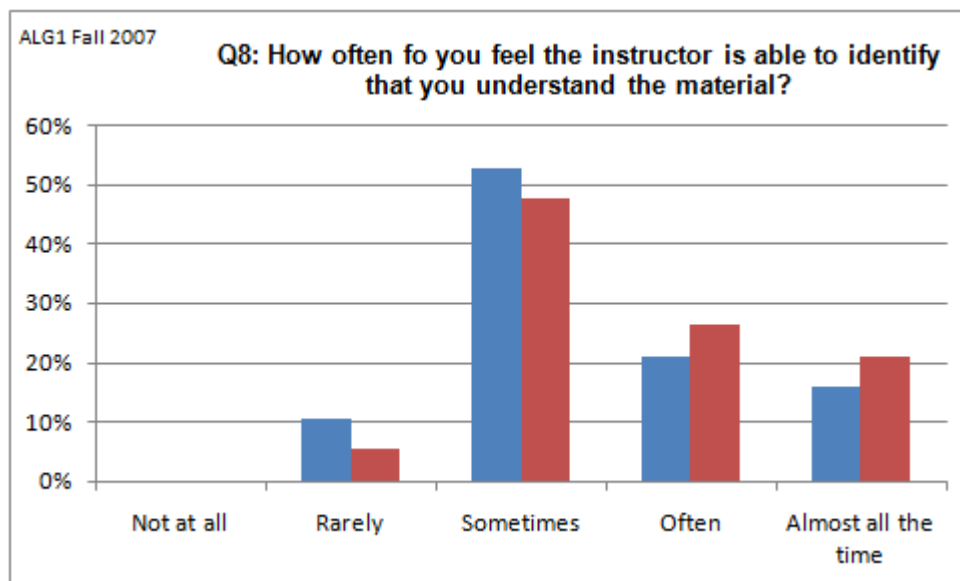


Figure A.50: How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

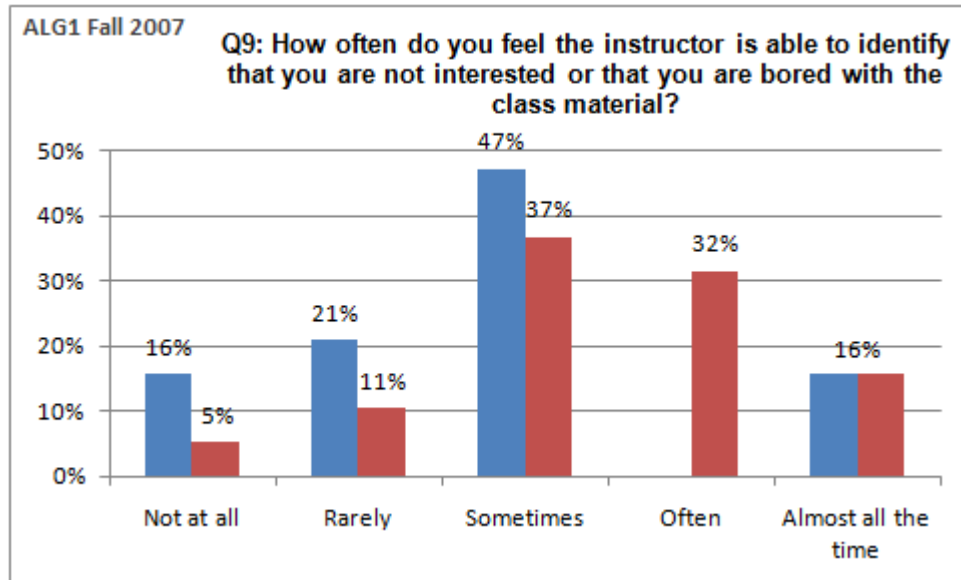


Figure A.51: Small group exercises are effective for improving my understanding of material.

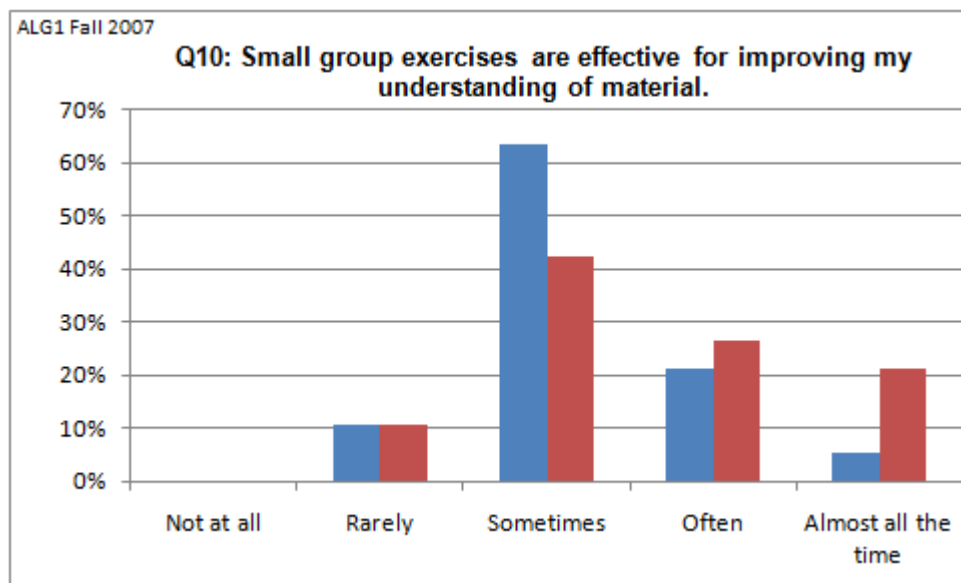


Figure A.52: The in-class activities keep me interested in the course content.

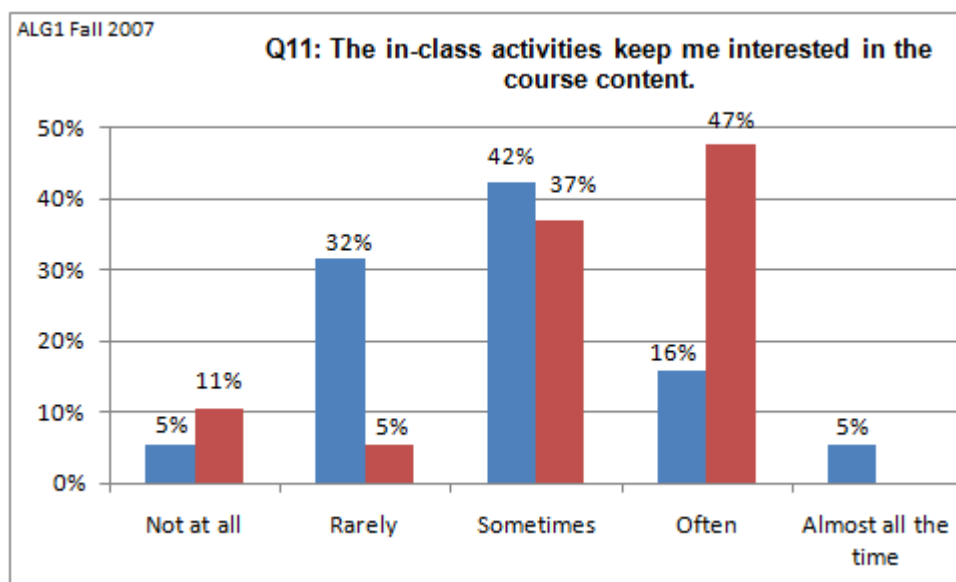


Figure A.53: My small group needs more instructor guidance to work effectively.

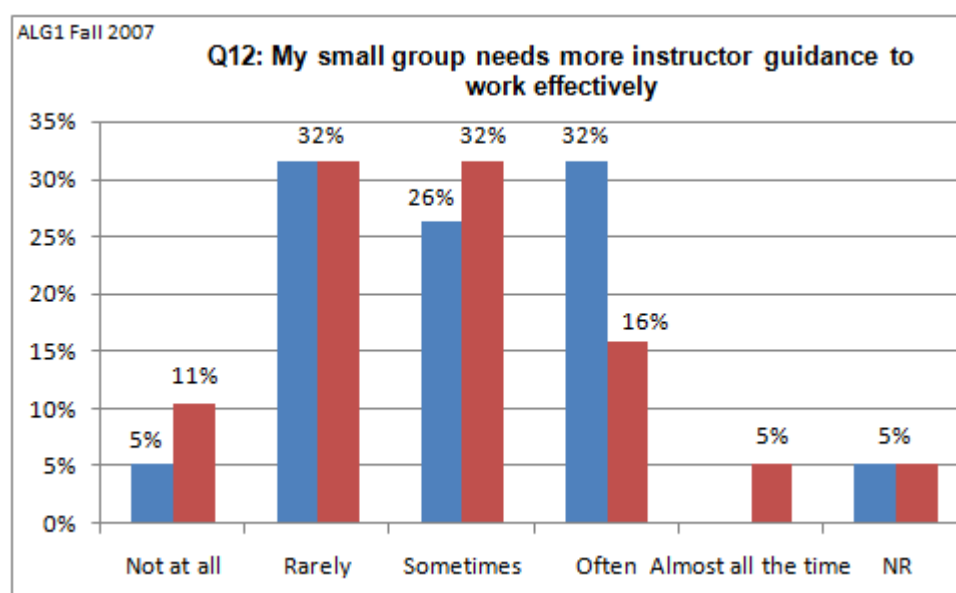
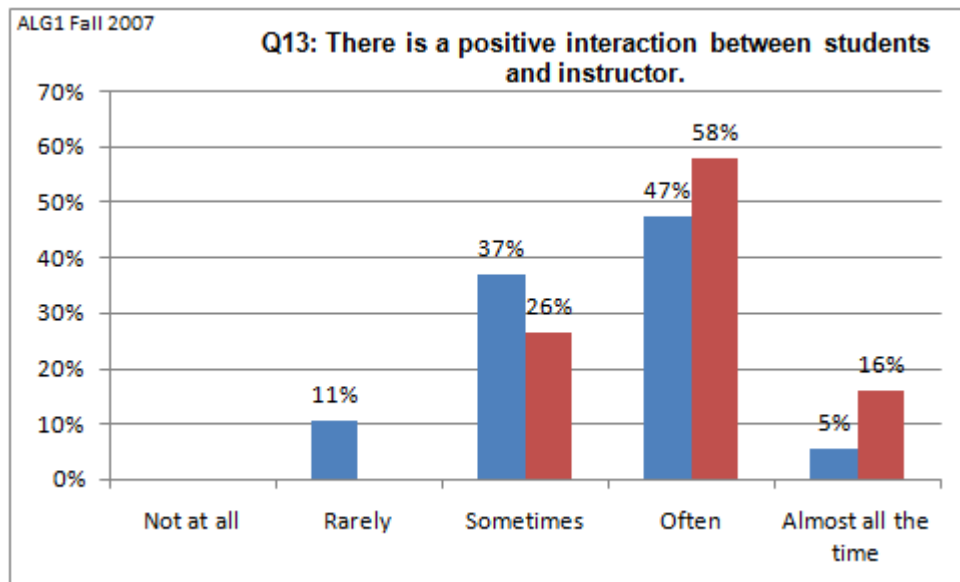


Figure A.54: There is a positive interaction between students and instructor.



Algebra 2

Figure A.55: The instructor promotes an atmosphere conducive to work and learning.

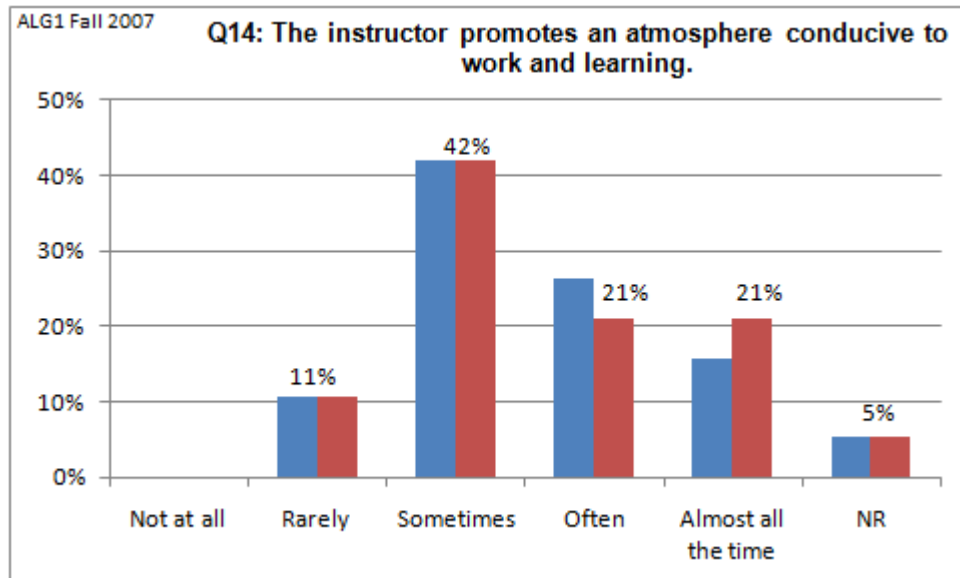


Figure A.56: The instructor promotes interaction amongst the students.

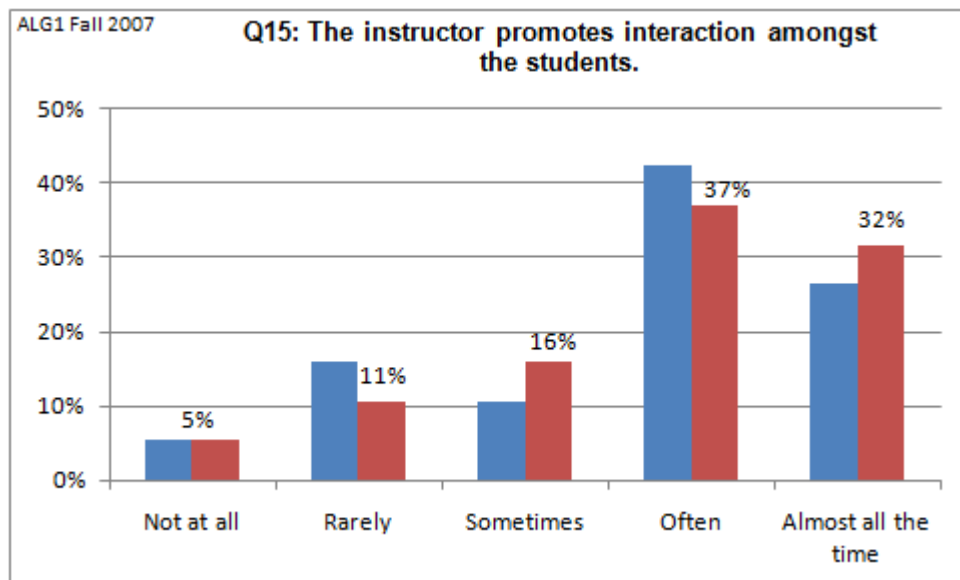


Figure A.57: The instructor encourages class members to work as a team.

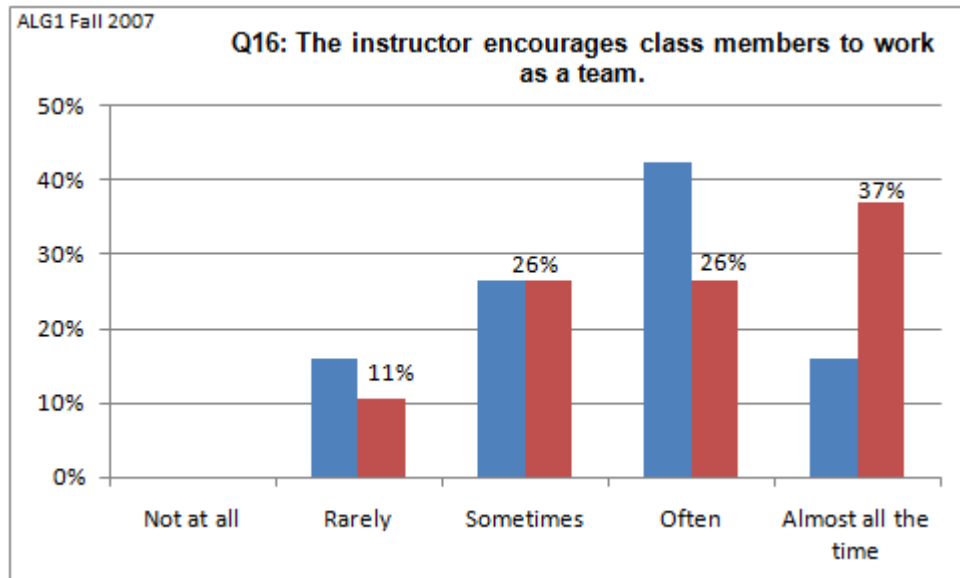


Figure A.58: How often do you seek out the teacher to ask a question outside of class?

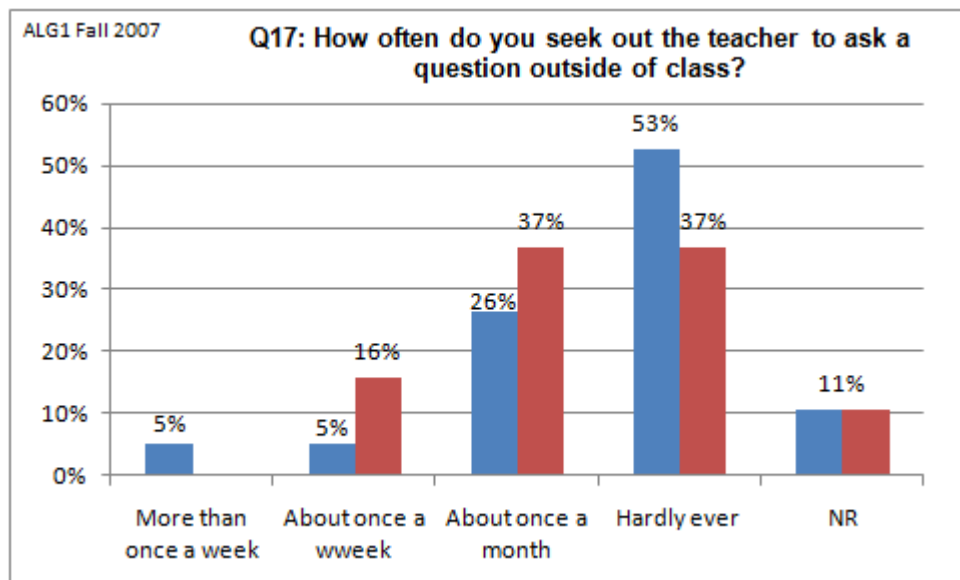


Figure A.59: Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them?

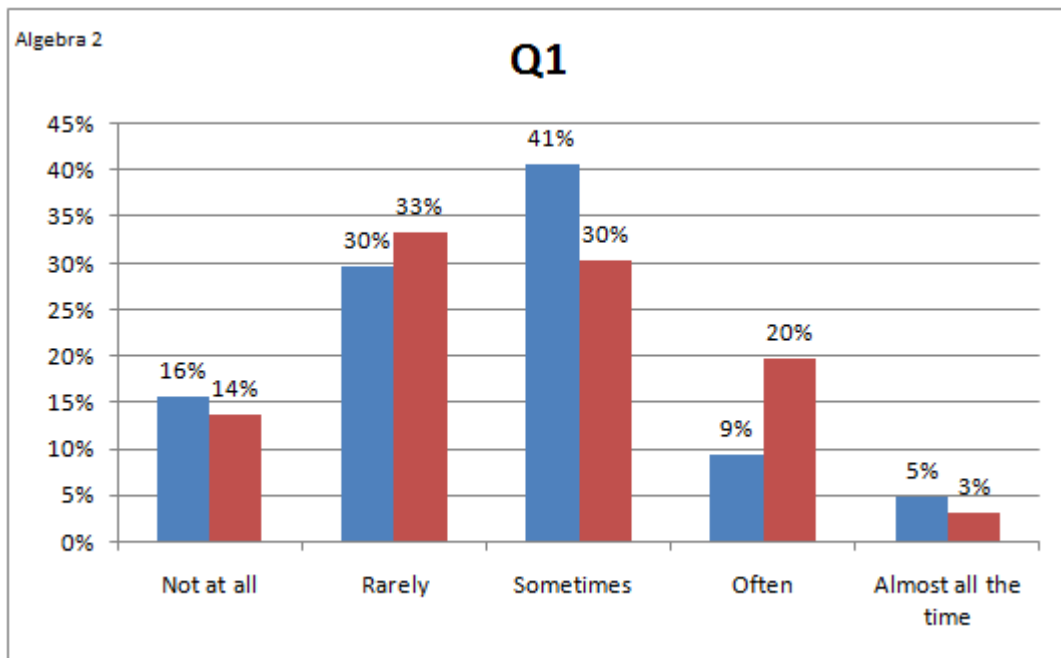


Figure A.60: In those cases when you have a question but don't ask it out loud, how often does another student ask it?

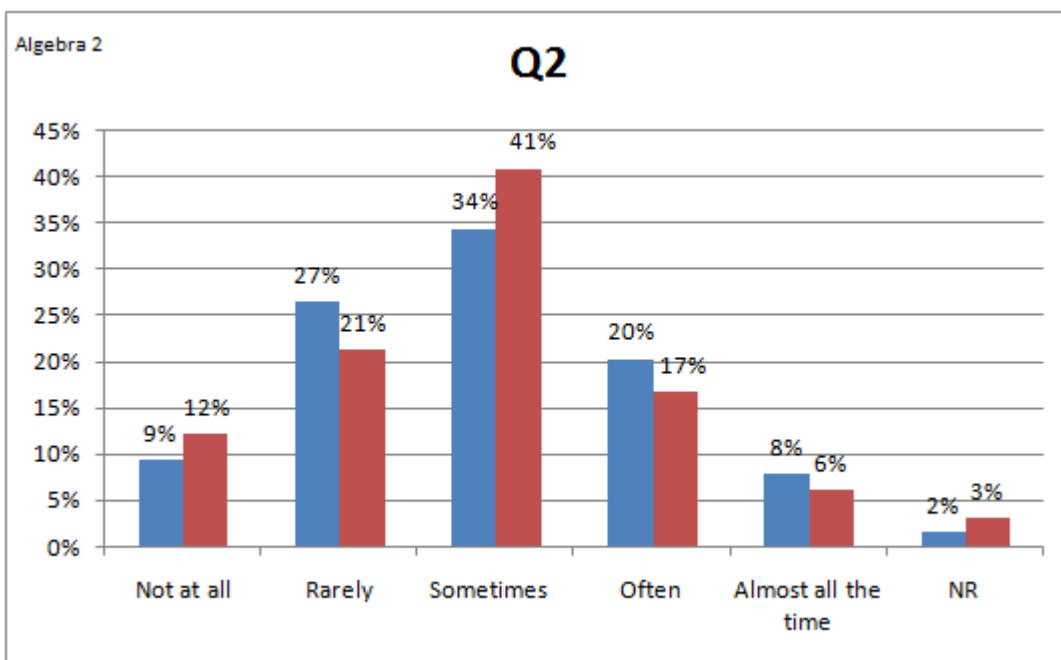


Figure A.61: When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking?

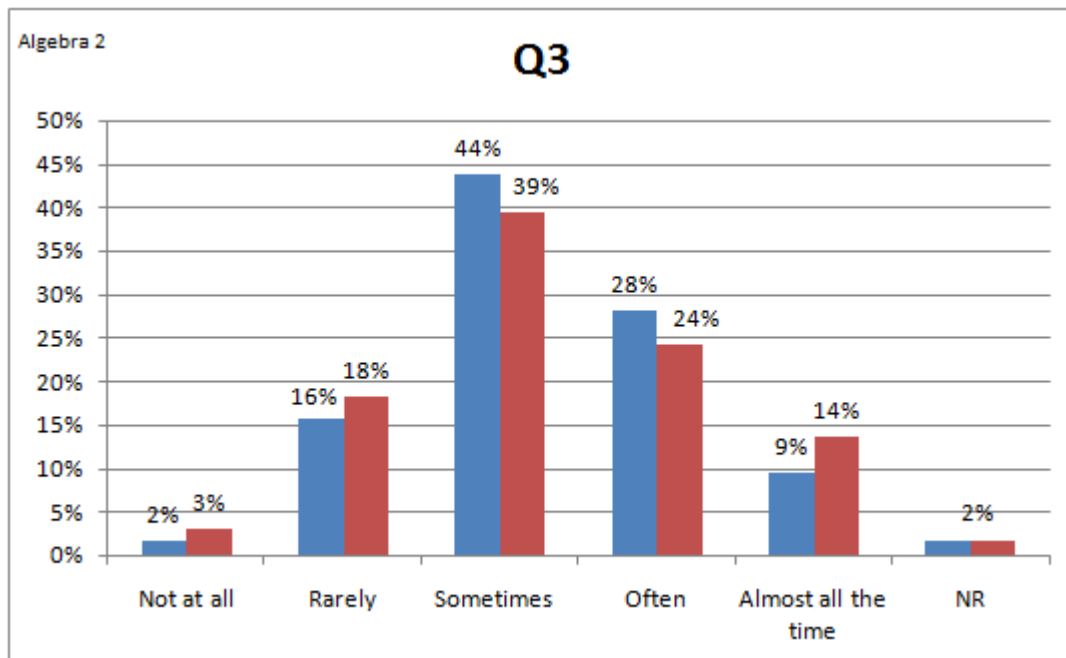


Figure A.62: How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)?

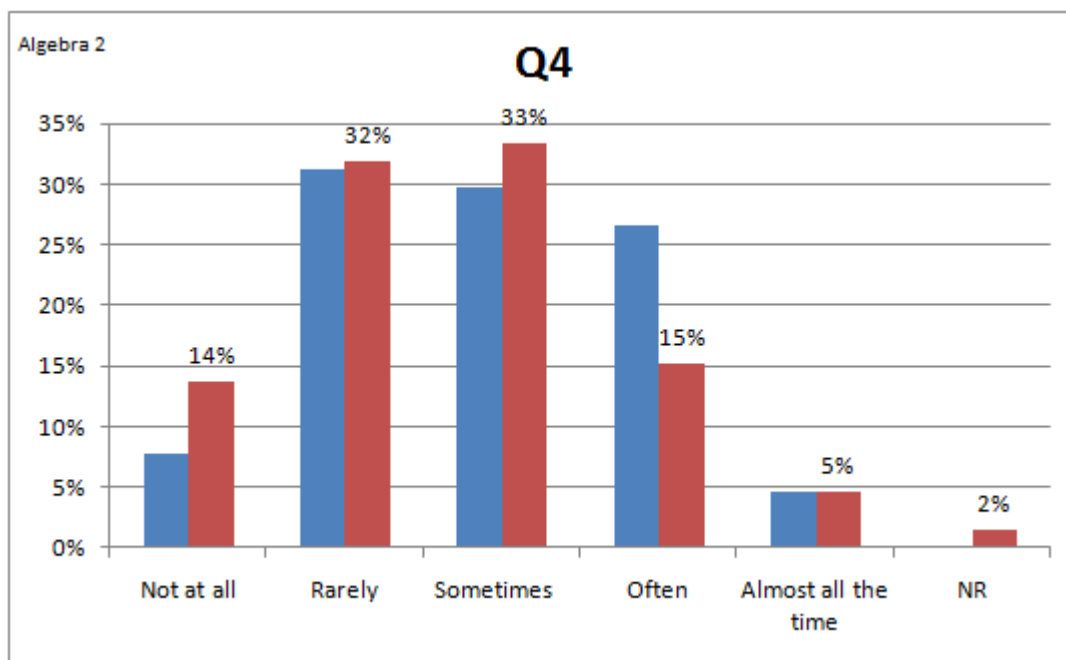


Figure A.63: How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)?

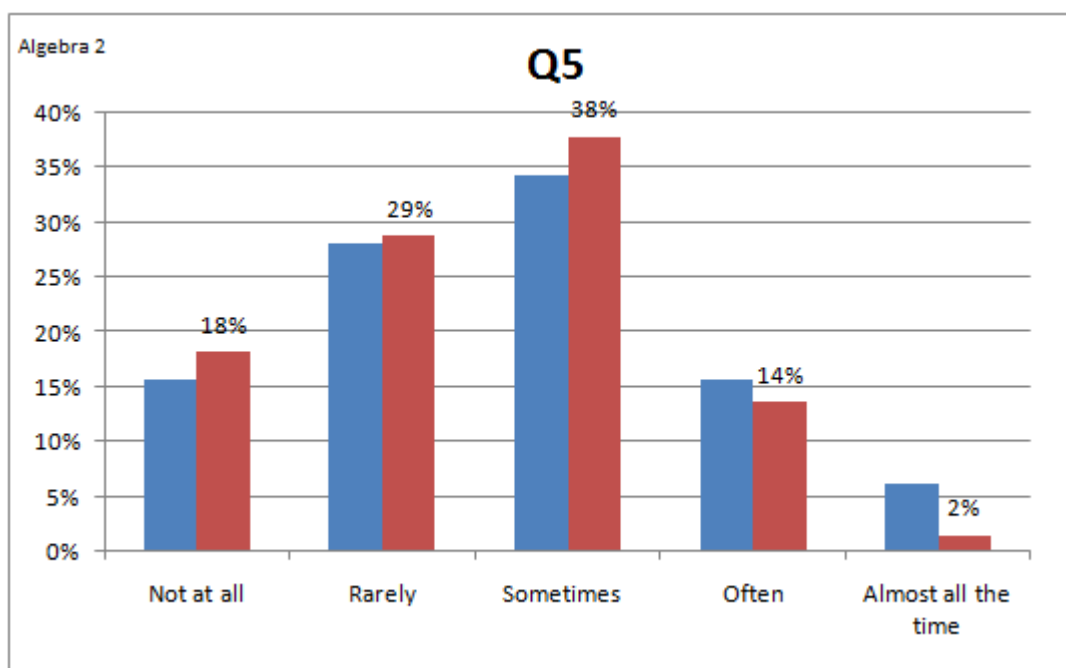


Figure A.64: How often do you feel the instructor offers you one on one attention during class?

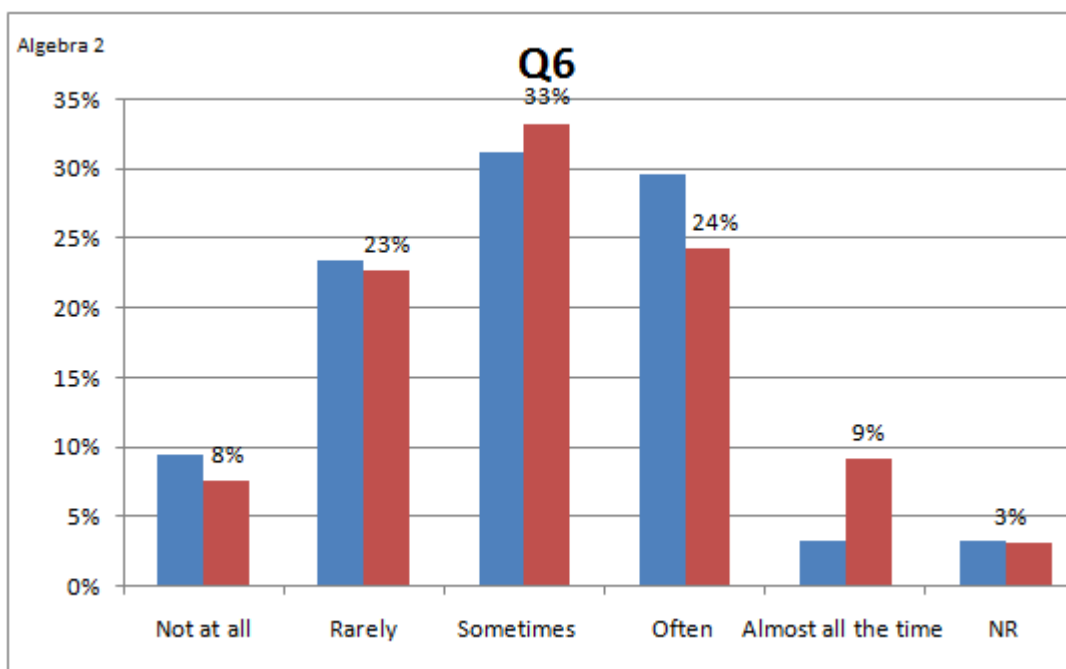


Figure A.65: How often do you feel the instructor is able to identify that you are struggling with a particular topic?

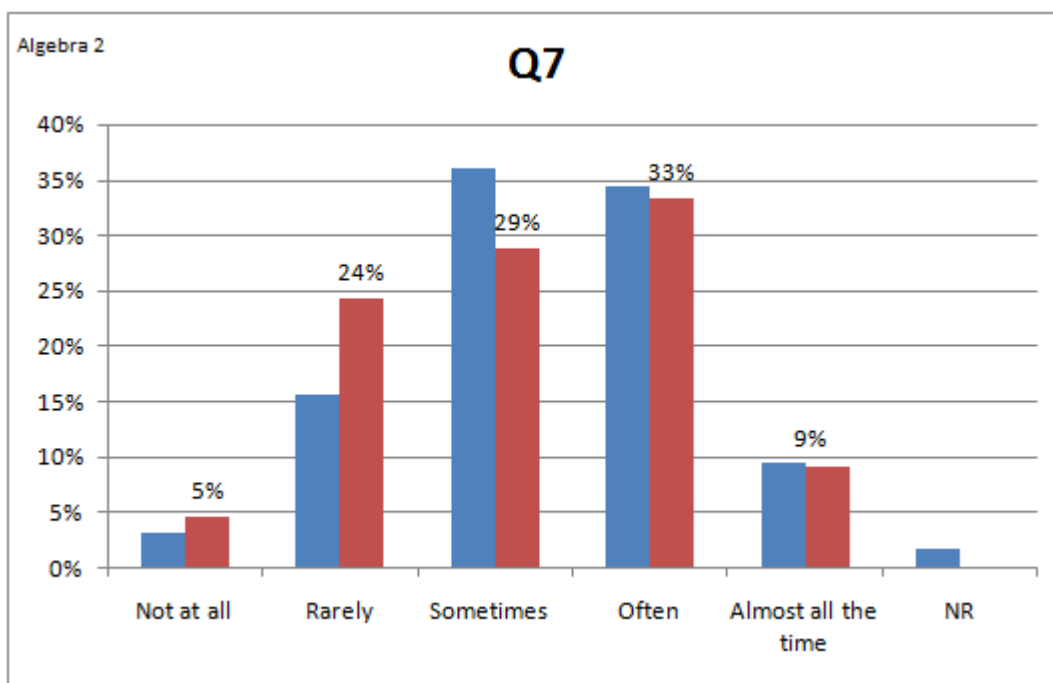


Figure A.66: How often do you feel the instructor is able to identify that you understand the material?

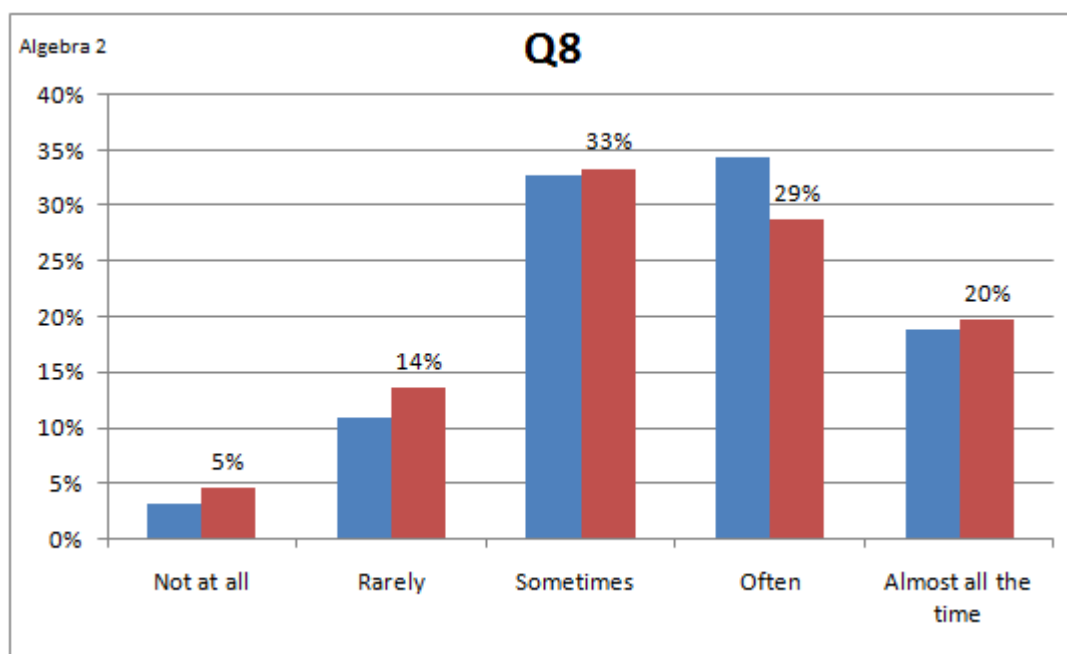


Figure A.67: How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

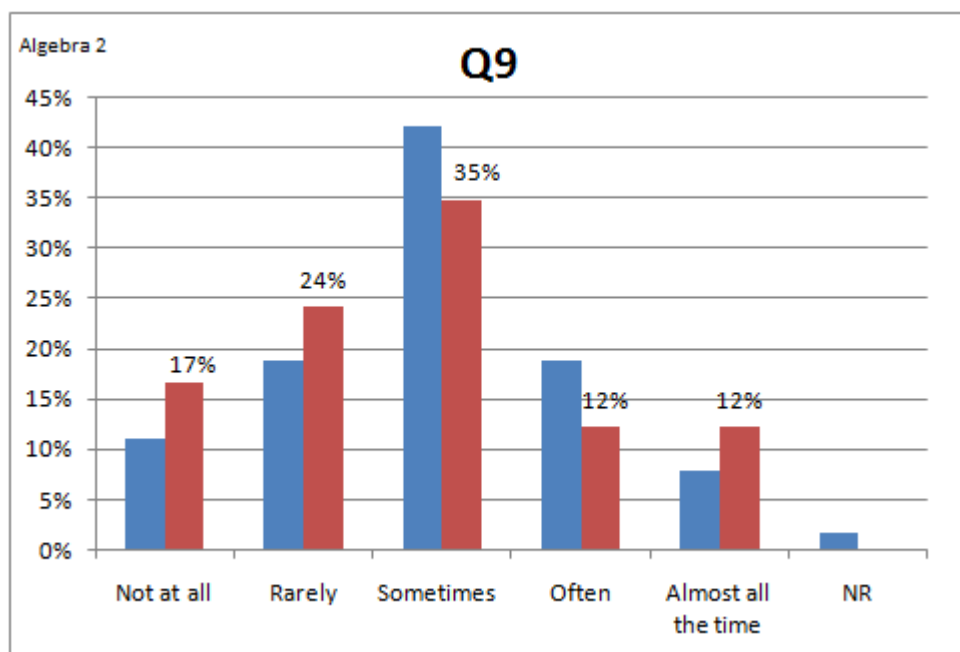


Figure A.68: Small group exercises are effective for improving my understanding of material.

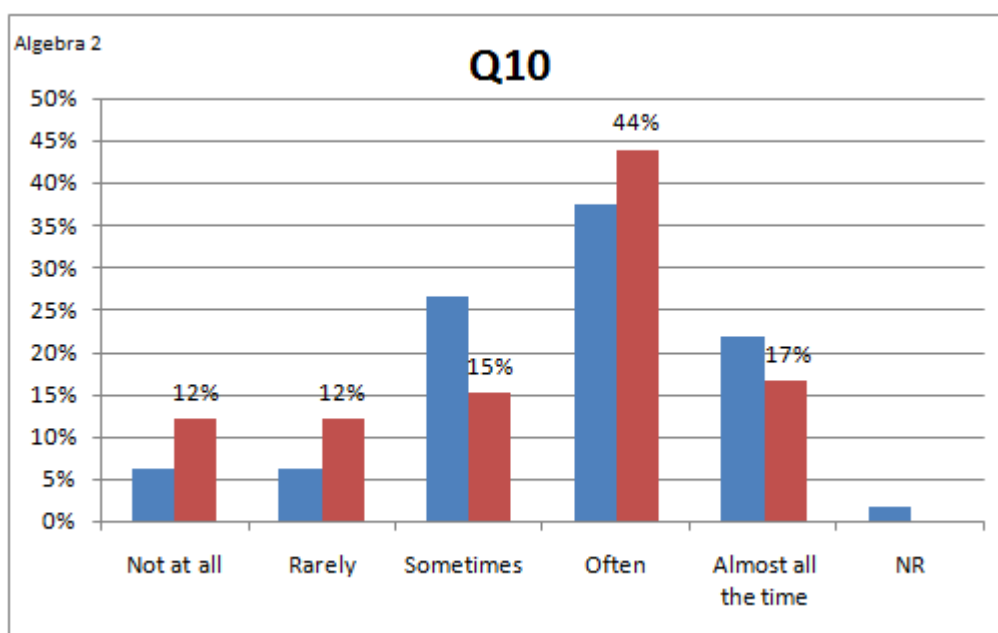


Figure A.69: The in-class activities keep me interested in the course content.

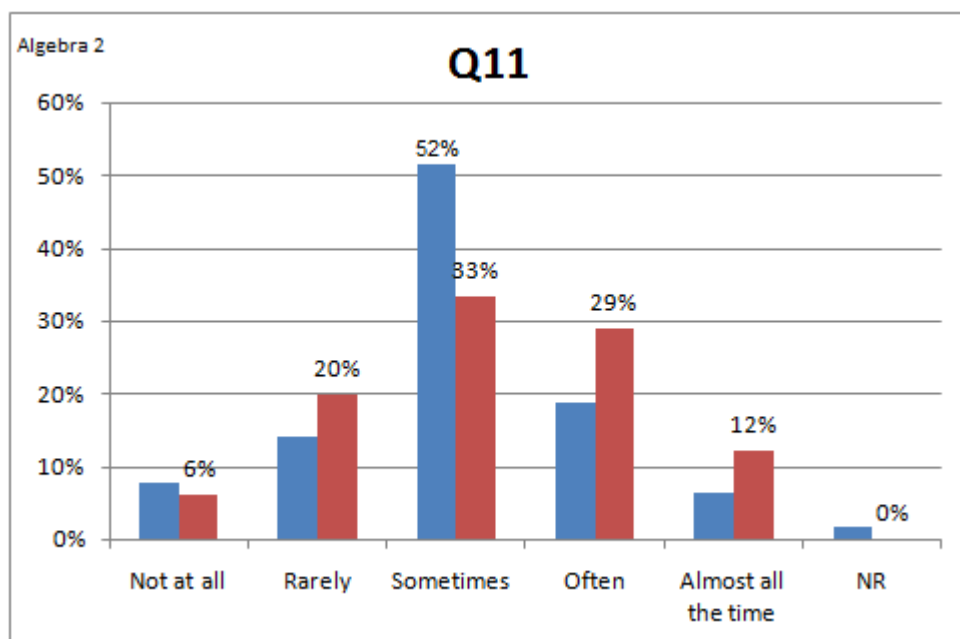


Figure A.70: My small group needs more instructor guidance to work effectively.

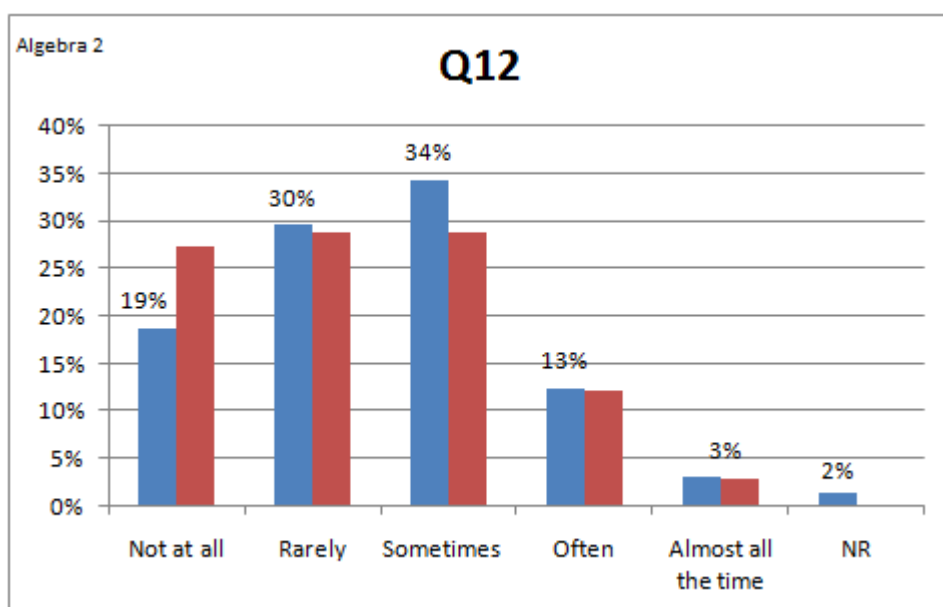
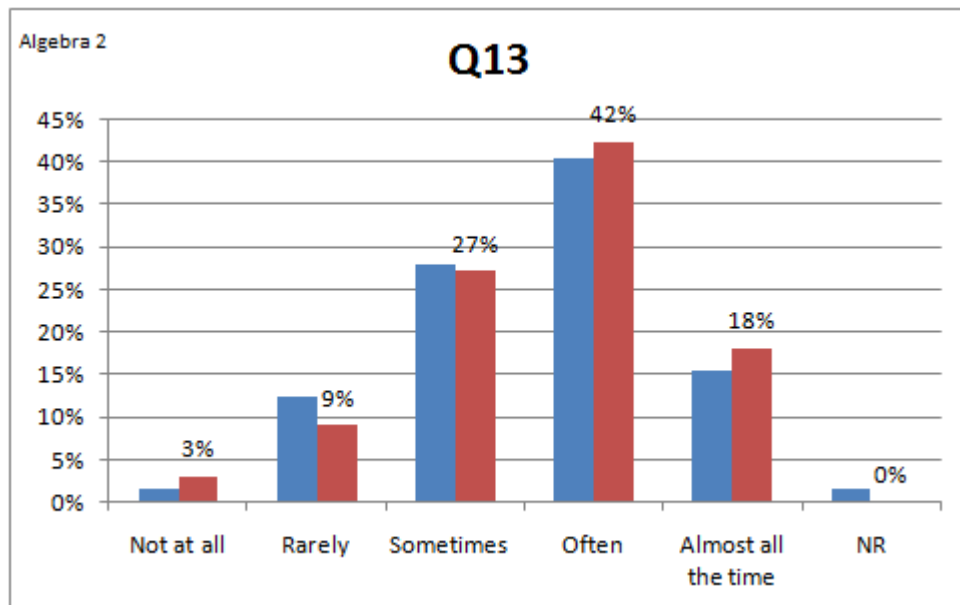


Figure A.71: There is a positive interaction between students and instructor.



Algebra 1 no technology

Figure A.72: The instructor promotes an atmosphere conducive to work and learning.

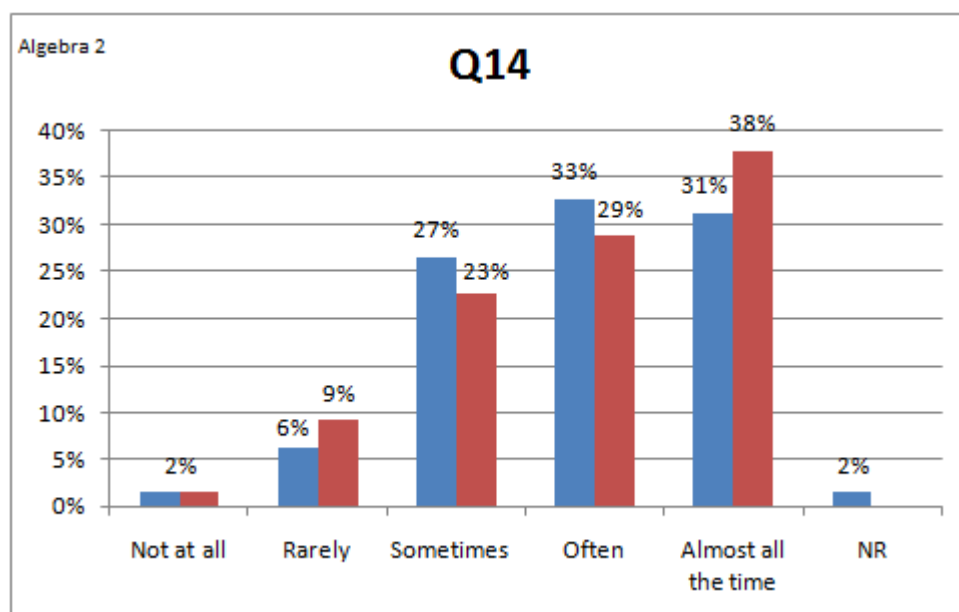


Figure A.73: The instructor promotes interaction amongst the students.

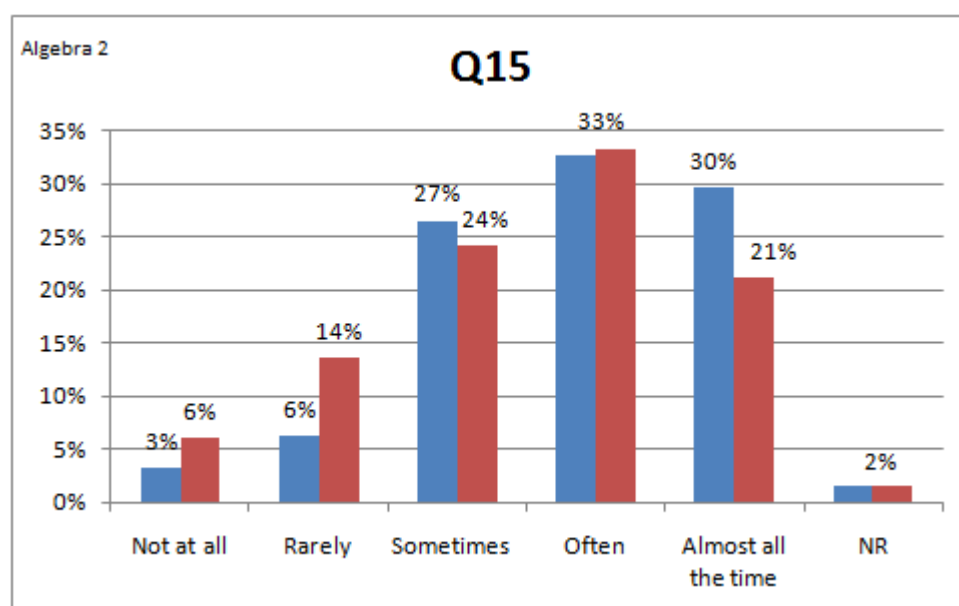


Figure A.74: The instructor encourages class members to work as a team.

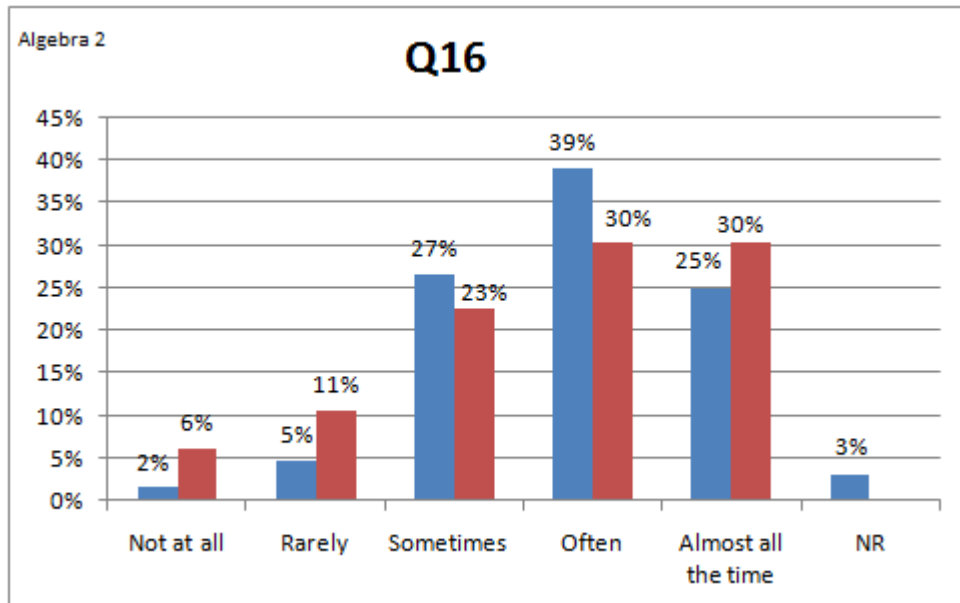


Figure A.75: How often do you seek out the teacher to ask a question outside of class?

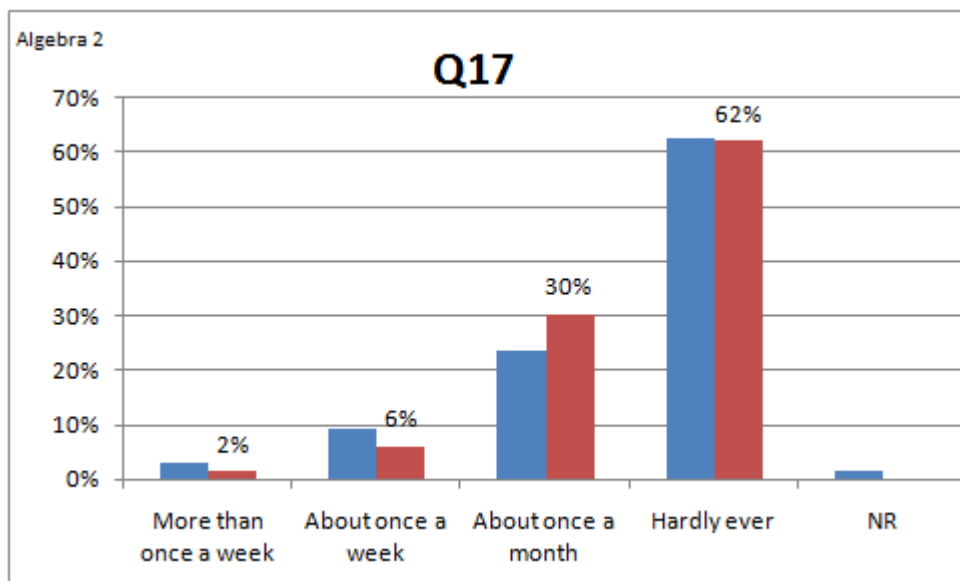


Figure A.76: Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them?

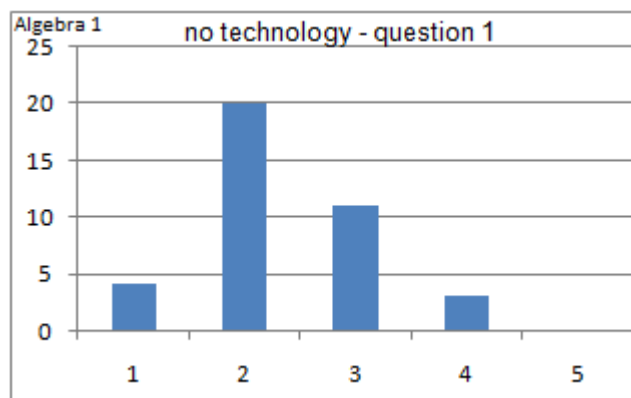


Figure A.77: In those cases when you have a question but don't ask it out loud, how often does another student ask it?

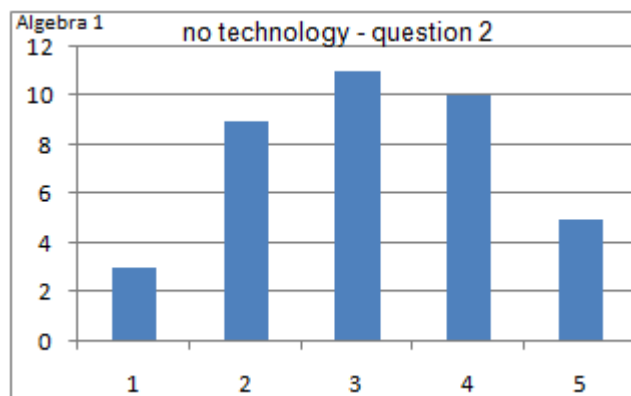


Figure A.78: When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking?

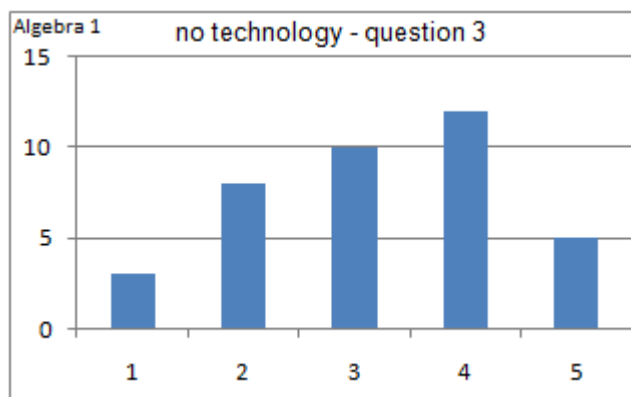


Figure A.79: How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)?

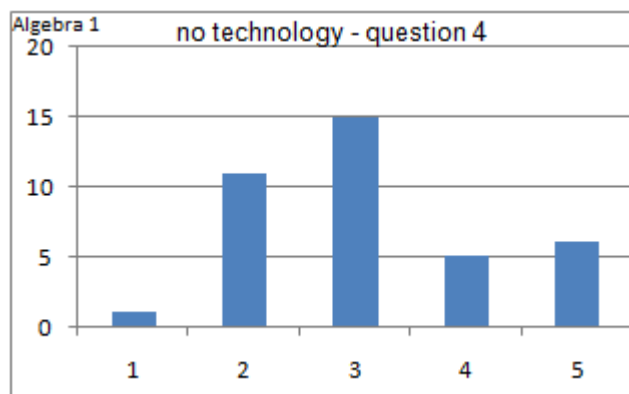


Figure A.80: How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)?

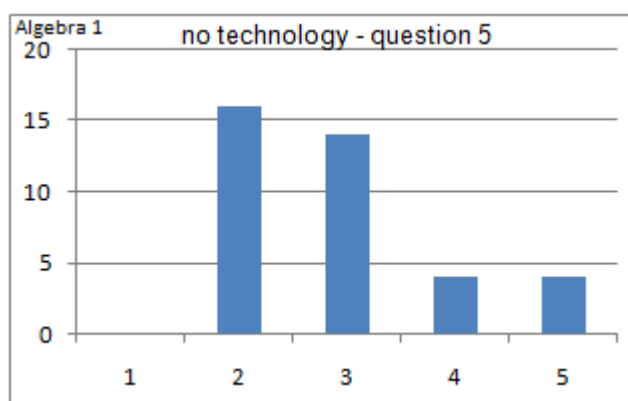


Figure A.81: How often do you feel the instructor offers you one on one attention during class?

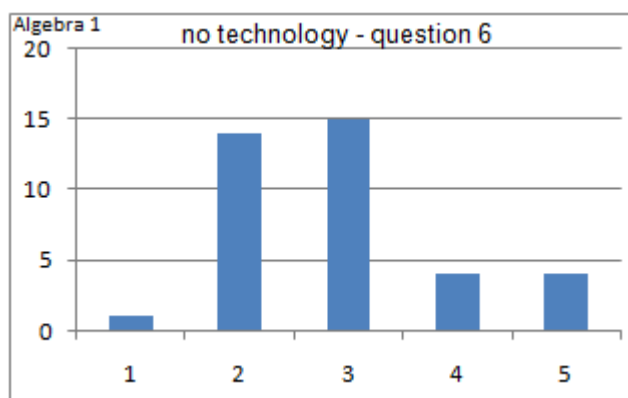


Figure A.82: How often do you feel the instructor is able to identify that you are struggling with a particular topic?

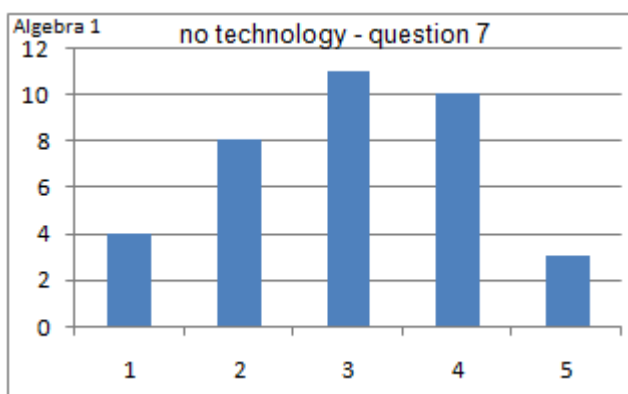


Figure A.83: How often do you feel the instructor is able to identify that you understand the material?

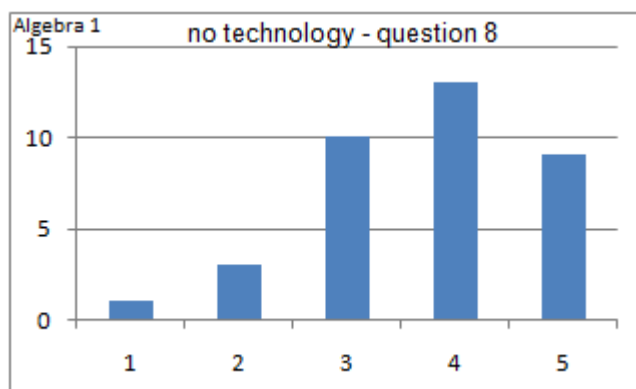


Figure A.84: How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

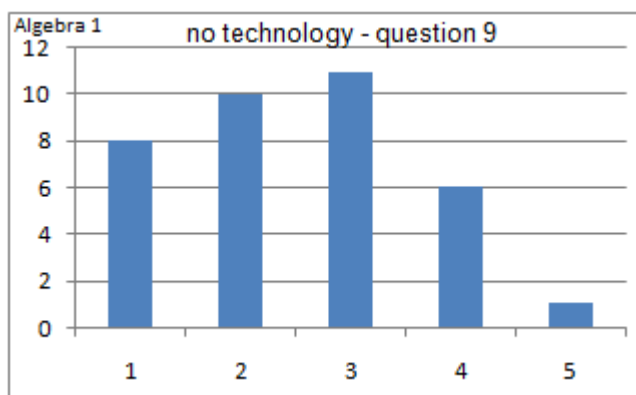


Figure A.85: Small group exercises are effective for improving my understanding of material.

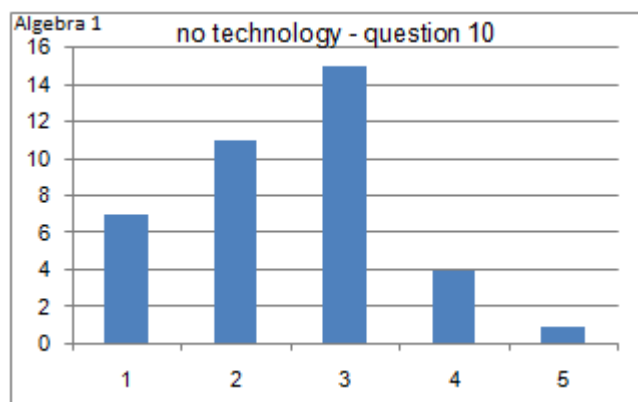
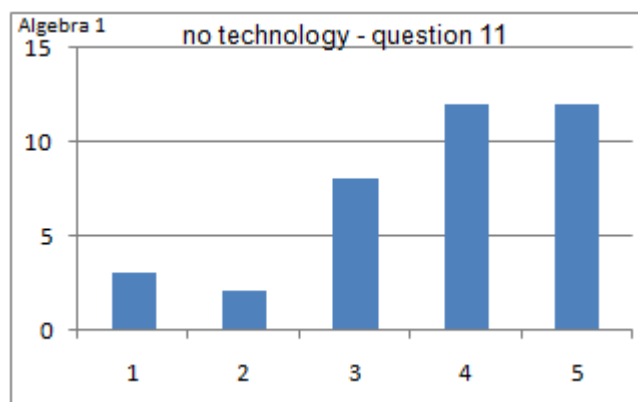


Figure A.86: The in-class activities keep me interested in the course content.



Geometry without technology

Figure A.87: My small group needs more instructor guidance to work effectively.

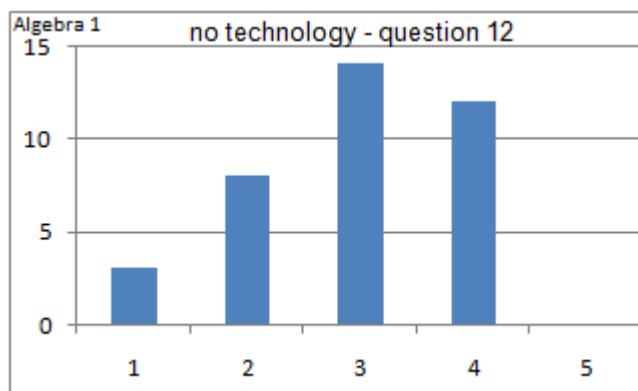


Figure A.88: Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them?

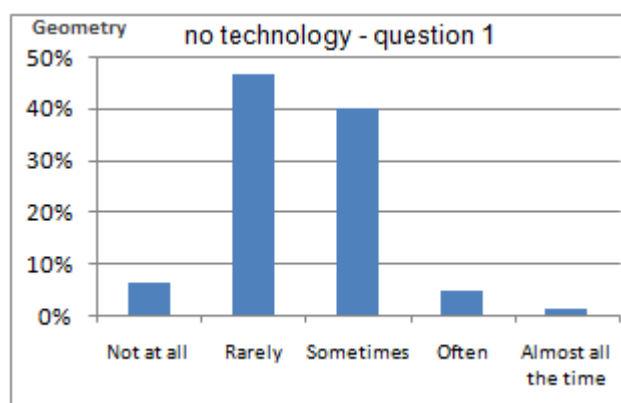


Figure A.89: In those cases when you have a question but don't ask it out loud, how often does another student ask it?

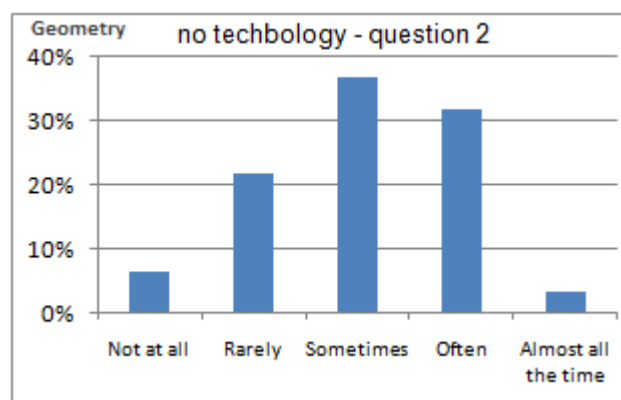


Figure A.90: When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking?

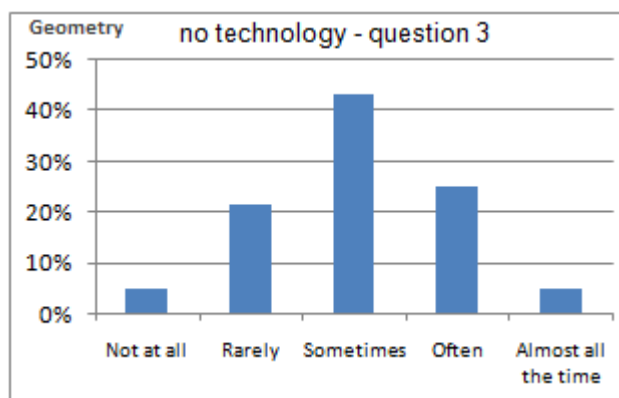


Figure A.91: How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)?

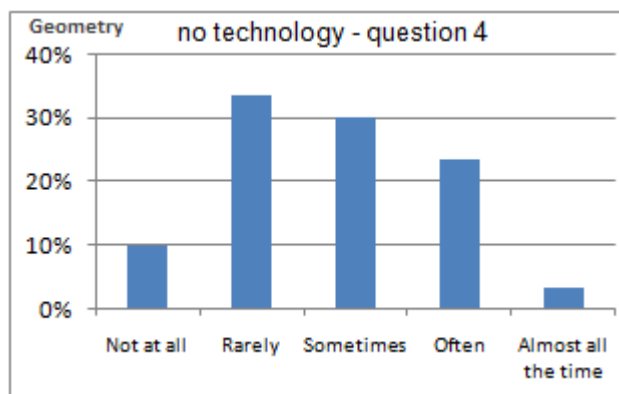


Figure A.92: How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)?

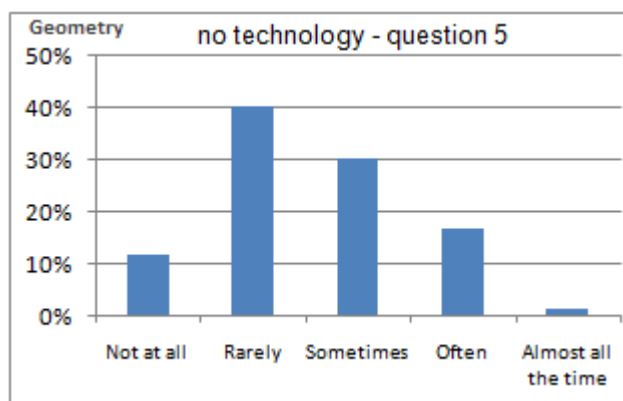


Figure A.93: How often do you feel the instructor offers you one on one attention during class?

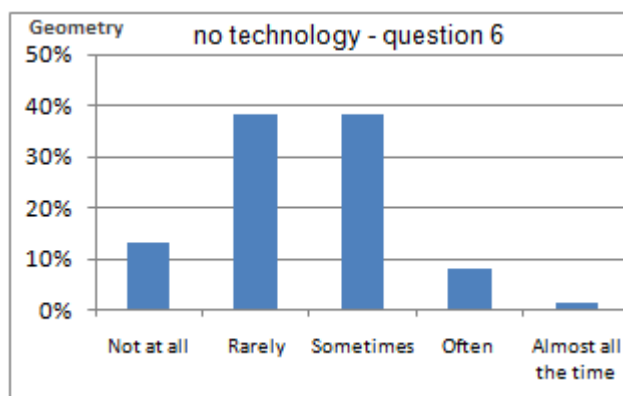


Figure A.94: How often do you feel the instructor is able to identify that you are struggling with a particular topic?

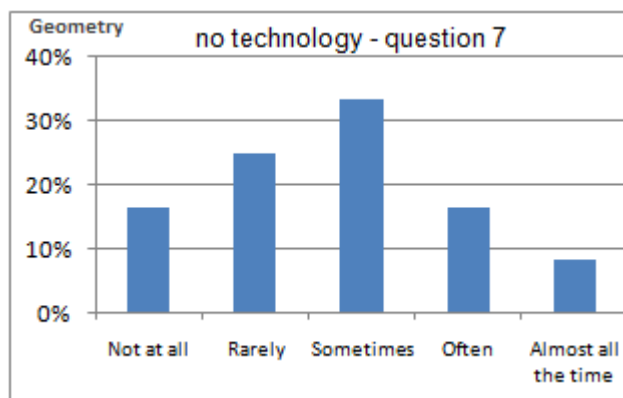


Figure A.95: How often do you feel the instructor is able to identify that you understand the material?

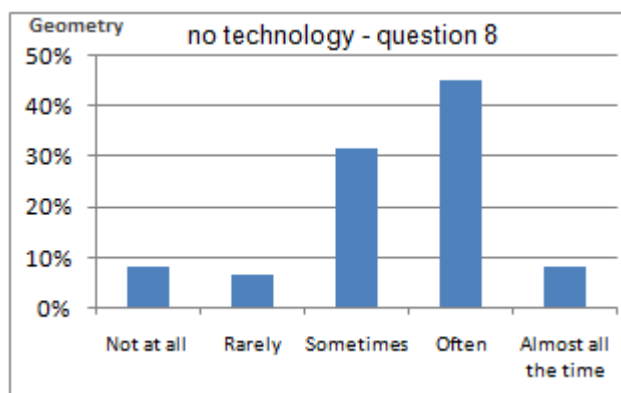


Figure A.96: How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

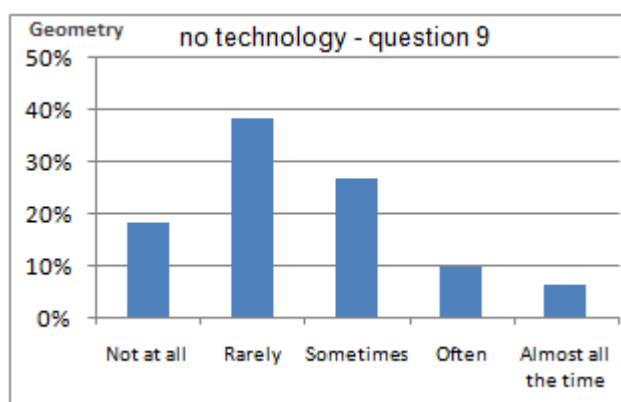


Figure A.97: Small group exercises are effective for improving my understanding of material.

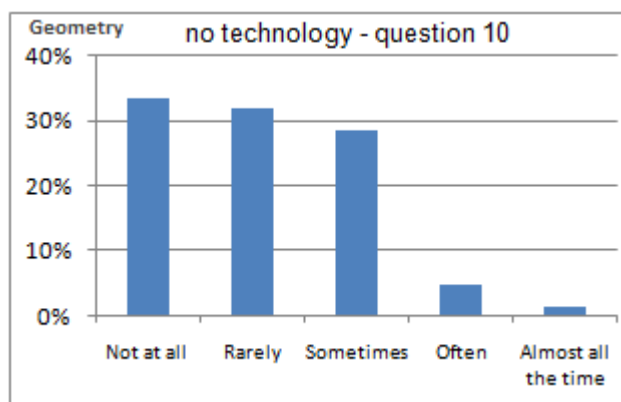
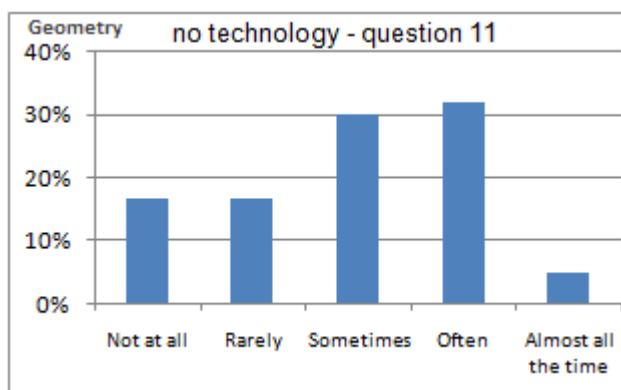


Figure A.98: The in-class activities keep me interested in the course content.



Geometry with technology

Figure A.99: My small group needs more instructor guidance to work effectively.

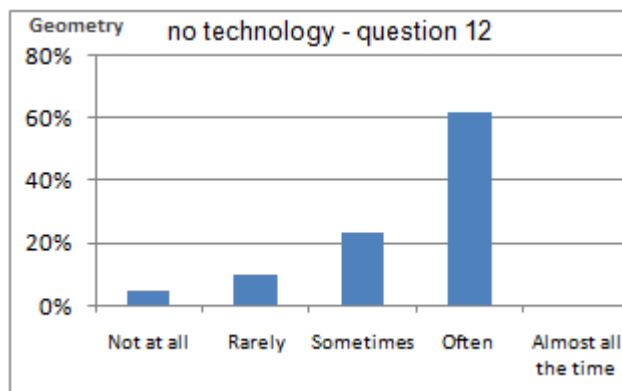


Figure A.100: Students are often reluctant to ask questions in class, because they are too shy, or they think the question is stupid or isn't important. How often do you have questions in class but don't ask them?

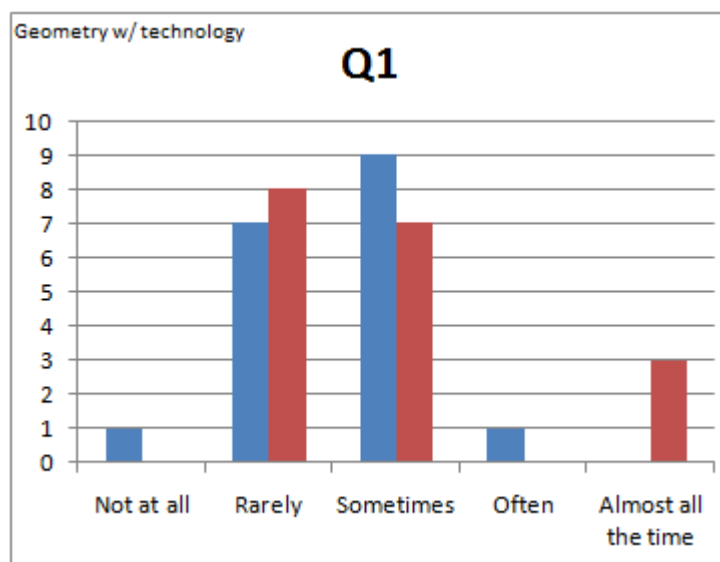


Figure A.101: In those cases when you have a question but don't ask it out loud, how often does another student ask it?

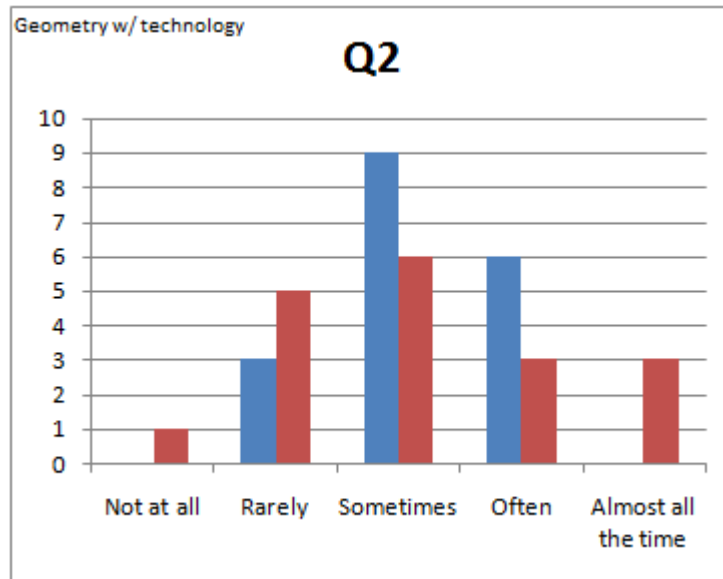


Figure A.102: When another student doesn't ask it, how often does the teacher answer it anyway? That is, how often does the teacher answer your questions without anyone asking?

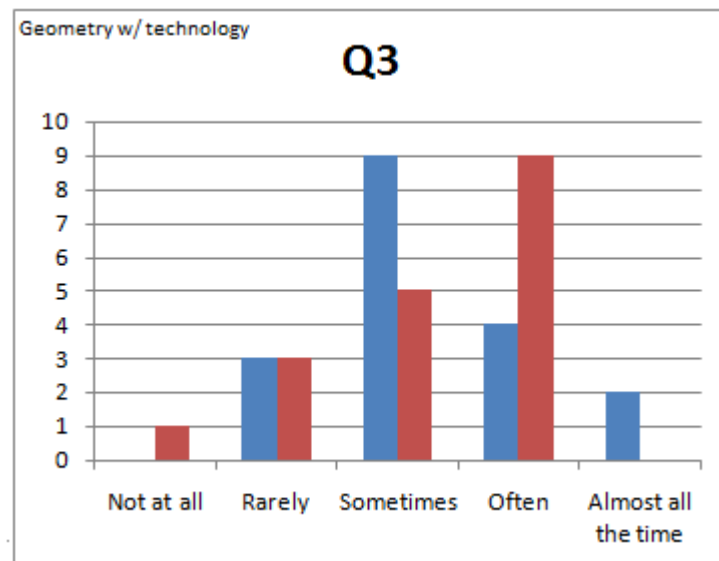


Figure A.103: How often do you discover misconceptions (misunderstandings) about the course material while doing homework (even when the material was covered in class)?

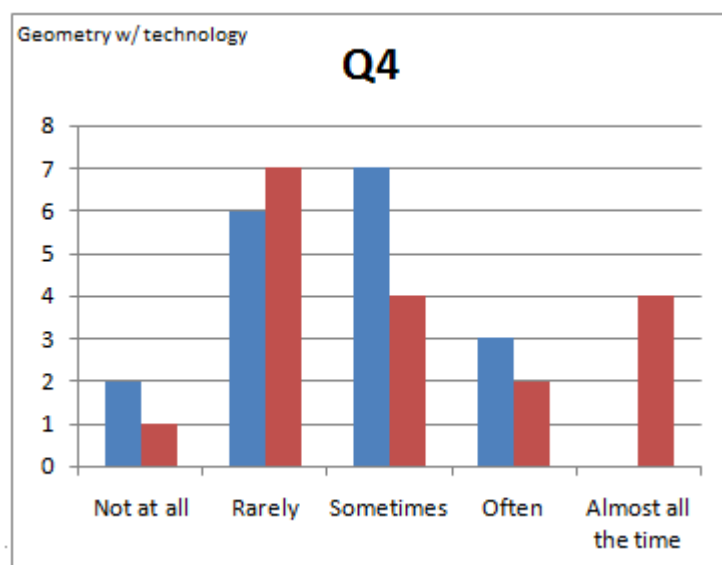


Figure A.104: How often do you discover misconceptions about the course material only after handing in your homework (either when the teacher goes over it in class or when the correct homework is handed back)?

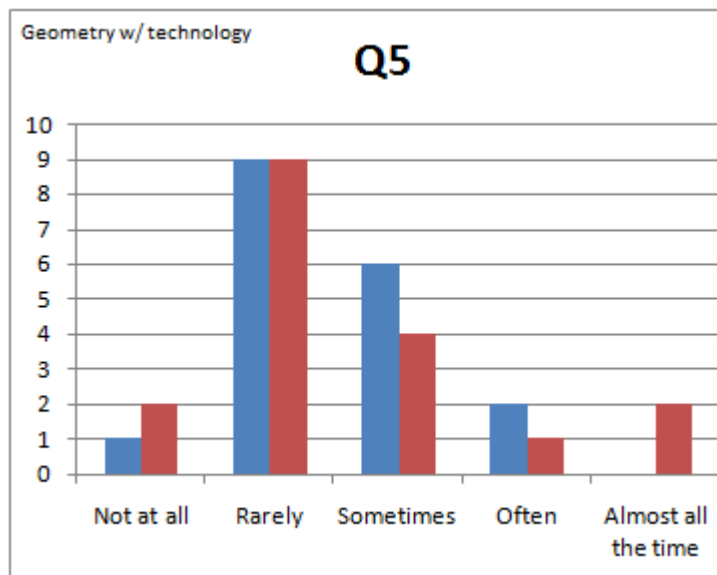


Figure A.105: How often do you feel the instructor offers you one on one attention during class?

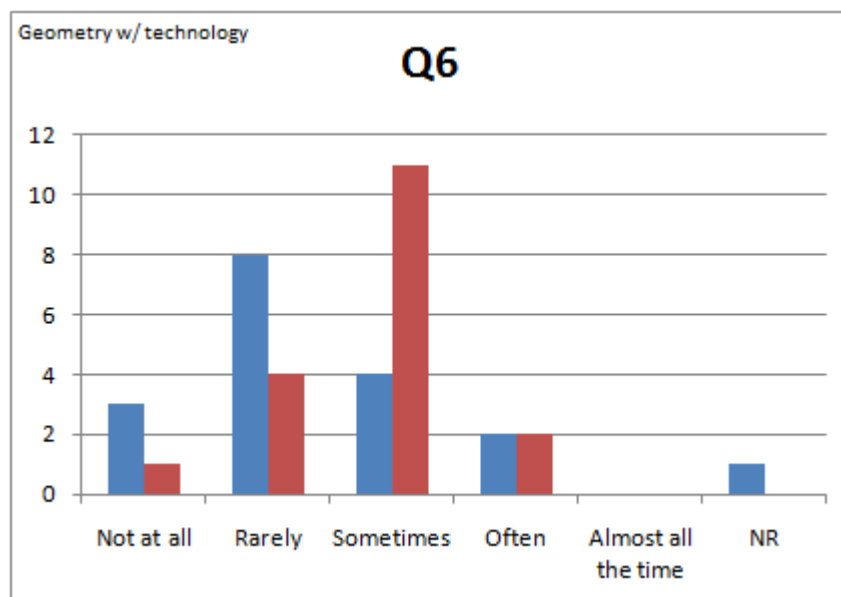


Figure A.106: How often do you feel the instructor is able to identify that you are struggling with a particular topic?

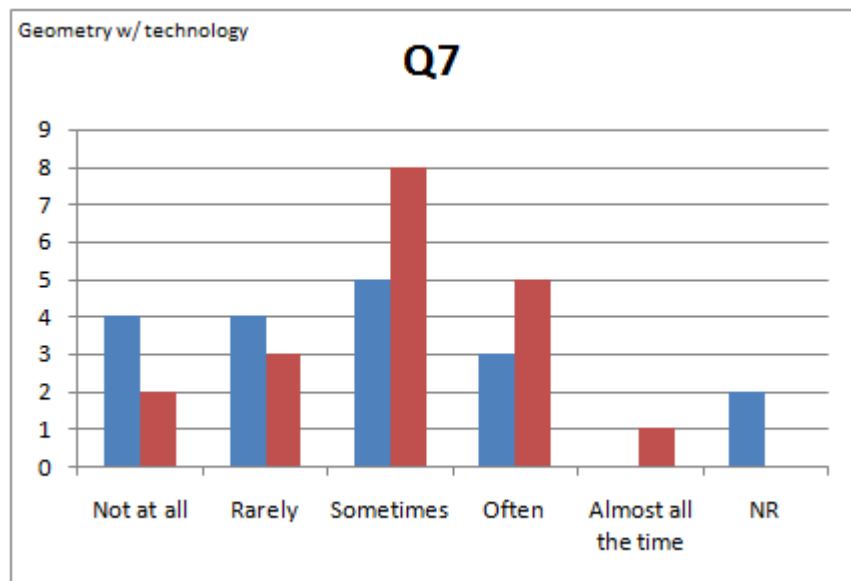


Figure A.107: How often do you feel the instructor is able to identify that you understand the material?

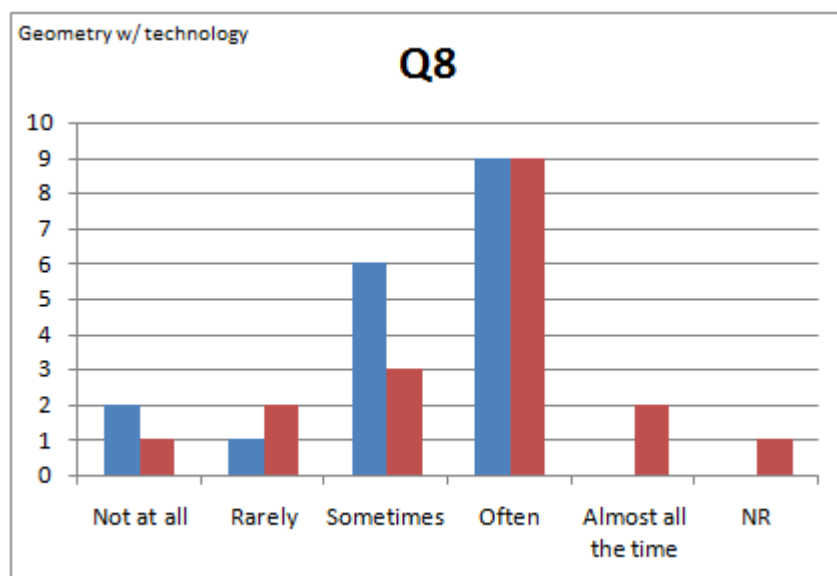


Figure A.108: How often do you feel the instructor is able to identify that you are not interested or that you are bored with the class material?

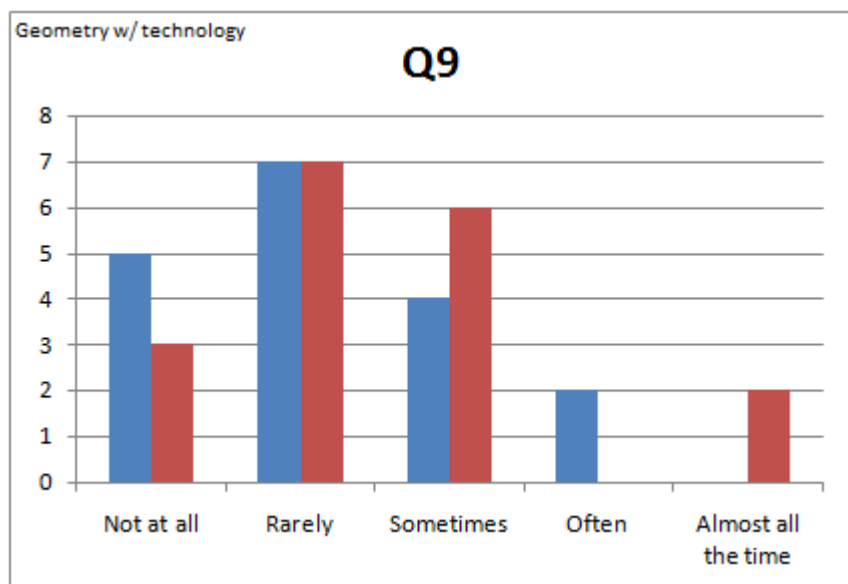


Figure A.109: Small group exercises are effective for improving my understanding of material.

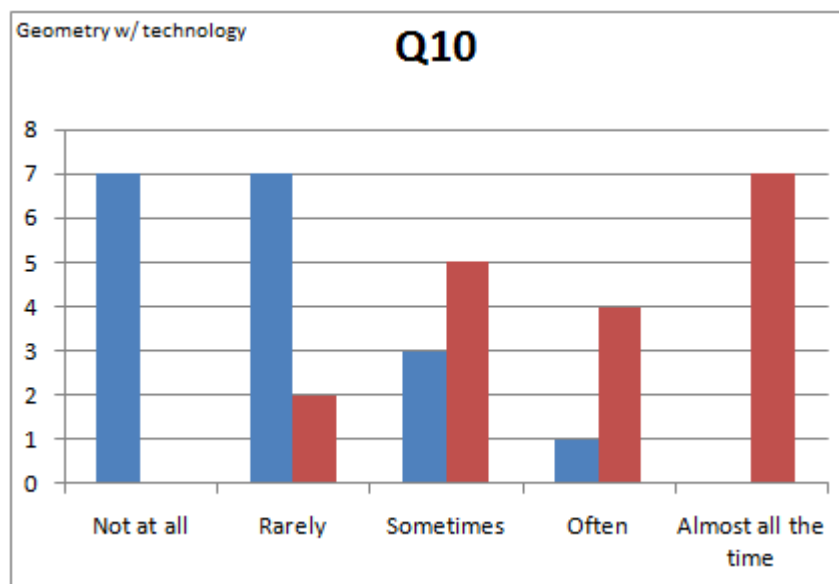


Figure A.110: The in-class activities keep me interested in the course content.

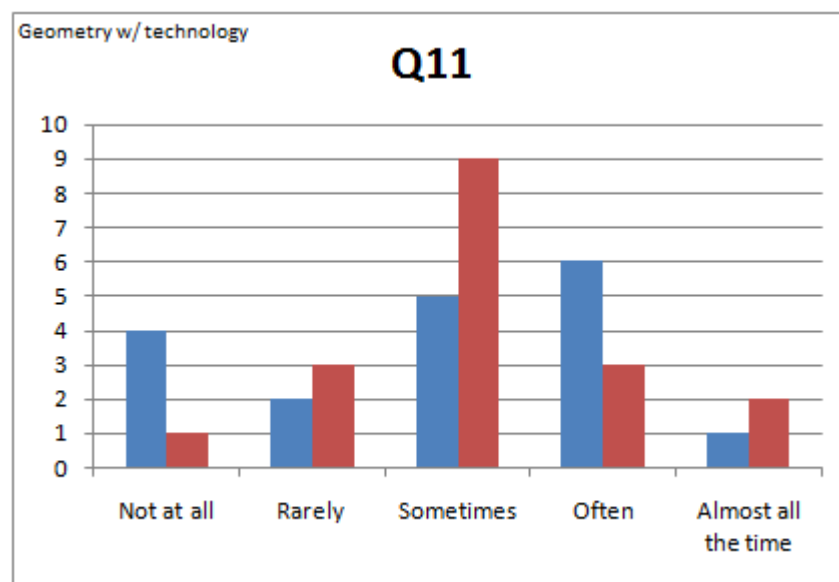
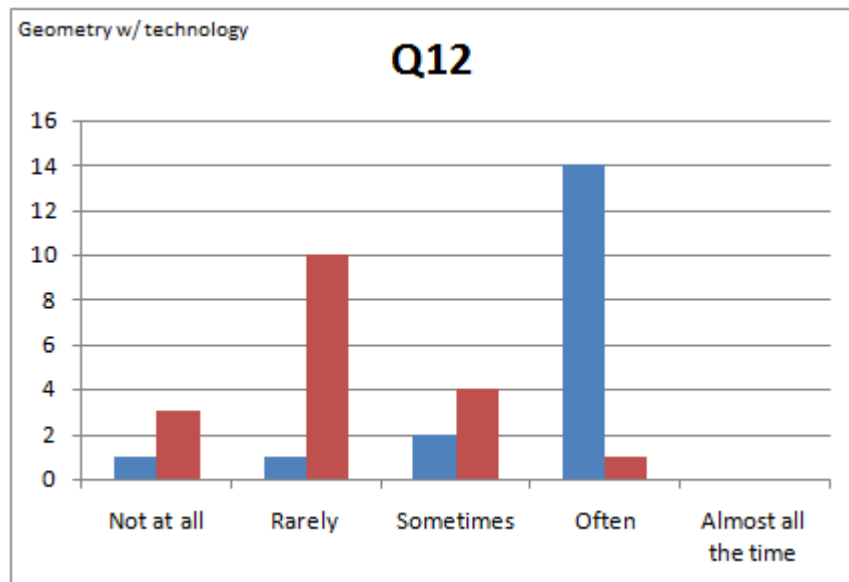


Figure A.111: My small group needs more instructor guidance to work effectively.



Appendix B

Instructor Reflections

B.1 Larry DeBrock, Instructor Reflections of Tablet PC Usage

I have switched to 100% utilization of a Tablet PC in my lectures and presentations. There is simply no comparison between the effectiveness of my lectures using the Tablet versus other delivery options.

Prior to using a Tablet PC, I tried to lecture using PowerPoint. This was a disaster on many fronts. First, using prepared-text slides tends to reduce students to robotic note-takers. The level of intellectual involvement was low. Second, the PowerPoint “bullet” format is not well designed to present the concepts in Economics. Third, it was difficult to put marks on the slides during lecture.

My solution was to turn to very old-style overhead projector technology. I would slap blank acetate sheets onto such a projector and mark them up with real ink pens. This was messy and wasteful, but it did solve the problems of PowerPoint mentioned above. However, it was not convenient as it was difficult to return to previously used pages during the lecture.

The solution, of course, is found in a Tablet PC with any of a number of whiteboard software applications. The Tablet PC allowed me to use the pedagogically superior ink markings rather than prepared text. And, it permitted me to make liberal use of markings, graphics, and highlighting through the pen. Finally, it offered a quick and efficient way to instantly return to a previous slide, complete with the markings.

I once had a professor in graduate school explain that classroom teaching made a great leap forward with the introduction of colored chalk and rooms with multiple chalkboards. The colored chalk allowed for better understanding of complex economics graphs/equations and the multiple boards meant the notes on each board stayed displayed for a longer time before they were erased to offer room to continue the lecture. A Tablet PC solves both of the problems quickly and in a much superior way, especially in larger classrooms.

B.2 Jason Zych, Instructor Reflections of Whiteboard Usage

Prior to using the electronic tablet, my lectures were delivered using one of two technologies – the whiteboard, and overhead transparencies. I liked both technologies due to the ability to write out examples as you gave a lecture, rather than being bound to pre-prepared material (as with Powerpoint, for example).

When I originally began using an electronic tablet, I saw it only as a replacement for the physical whiteboard. The whiteboard had a number of flaws, all of which were corrected by the tablet:

1. Words and drawings written on the whiteboard were difficult to see from the back of a large lecture hall; the electronic tablet could be projected onto an overhead screen for improved visibility
2. Whiteboard markers tended to fade rather quickly, which again interfered with visibility from seats further back in the lecture hall. The colors of the electronic tablet were never faded.
3. Whiteboards tended to get dirtier over time, again making writing hard to see. The electronic tablet always had a “clean” (pure white) writing space.
4. More colors were available on the electronic tablet, which was useful in some situations.
5. Certain standard shapes were provided by the tablet whiteboard programs available for use. These standard shapes could be used to quickly create a set of well-drawn nodes (for example).

However, as I became more comfortable with the use of the tablet, I discovered that tablets carried other advantages with them as well.

1. The whiteboard had been rather wide – meaning you could write things on one section of the whiteboard, move on to lecturing on the second section of the whiteboard, and yet refer visually to the older writings on the first section. The electronic tablet had a much smaller viewing area, so you needed to create new “pages” for more writing, and scroll back and forth between the pages to show students older or newer writing. I originally saw this as a disadvantage of the electronic tablet, but over time, I realized that it was really a mixed bag. While I could not show as much at one time as I could on the regular whiteboard, I *was* able to keep the entire lecture in the computer memory, whereas with the physical whiteboard, I’d eventually have to erase something permanently. If, for example, at the end of the lecture, I wanted to scroll the students through the highlights of the lecture again, that could be done with the tablet but could not have been done on the physical whiteboard.
2. The whiteboard programs of the electronic tables provided traditional computer-editing features such as copy-and-paste and undo/redo, as well as pre-programmed shapes. The copy-and-paste feature was

especially important. Combined with the previously-mentioned “paging” feature, it became possible to create “snapshots” of a diagram in time. When drawing a diagram, tracing through an algorithm step-by-step, etc., when I reached a point that I wanted to save as a snapshot, I could simply copy and paste the existing diagram onto the next page and continue editing in on the next page. The older version of the diagram, still on the previous page, could be referred to whenever I wanted to show a “work in progress”

This gave the benefits of “time-lapse” drawings that you can get from pre-prepared slides, while still being able to develop those drawings in real-time during lecture. On the physical whiteboards, you could only get a second copy of an existing diagram by actually drawing it a second time.

3. I was able to save my writings during lecture, export them into externally-viewable formats (such as PDF), and post them for students to download. The regular whiteboards we used did not offer this feature – to gain the ability to print out one’s writings, we would have needed to invest in the newer whiteboards that had printers built in, and even then, we would have had to make color copies of the printouts for them to be as useful to the students as the downloadable PDFs we could put on the website.
4. Once I moved from the NetMeeting whiteboard to a custom whiteboard program developed at UIUC, I had additional standard shapes to use, if I desired to use them.

B.3 Craig Russell, Algebra 2 Instructor

1. “Did the technology help you (the class, the students) in any way? How? Technology refers to the Tablet PC, the dashboard (the final version with the sorting), the application the students used, and even the SVN web repository.”

The tablet PC was useful, because it was portable and easy to use. Grading electronic exams using the tablet (with NuPaper® software) was particularly easy. Drawbacks included not having the same software on every platform I used and occasional network connectivity issues.

Dashboard was useful, and would have been even more so in a larger class.

I did not use most of the sorting features (mostly sort by group or sort by page number). The way I used the dashboard, primarily, was to monitor work that was ongoing in groups; I could sometimes catch errors as they occurred and intervene, and, knowing which students and groups to watch based on my experiences with the students, I was able to get visual cues and give feedback rapidly where it was needed most. Occasionally getting screen shots of the dashboard proved useful, too, because I could see at a glimpse how far everyone had gotten.

The student use of NuPaper was a bit puzzling to me at first, and and still remains so. The interaction between NuPaper and the dashboard being what it was, I suppose it was necessary to have students “ink” on their tablets in order to see their work live, but it seems to me that NuPaper was an extremely limited use of available technology. Students had to use calculators, for instance, instead of computer software; students had to sketch by hand rather than using drawing software. Of course, if my class had been in a regular classroom, students would have had to use calculators and draw by hand... but those are calculations and graphs are fairly common uses for computers, and students couldn’t use the computers for those functions. Of course, I understand the purpose of the study was to examine teacher-student interaction pedagogy using computers, but with mathematics, more and more it should be a teacher-student-technology interaction that shapes pedagogy. There really was no possibility for interactive technology with NuPaper as I saw it used.

I really liked the concept of the SVN repository, but I never used all the “versioning” features, and some functions (especially toward the end of the school year) didn’t work as they had earlier. I would like to have had a little more practice in using it—such as setting up file structures for all student files at once, instead of in individual files. The file naming protocol we used was unwieldy for students; they might have found a sequential outline file numbering system more useful than a rather long string of seemingly random text followed by a date. Some students used the repository, frequently, but many

never used it.

2. “I know in the beginning you spent a great deal of time creating worksheets that we thought would be more conducive to the Tablet PC (like the addition of answer boxes, etc..) How did this change over time?”

Over time, I became more adept at modifying existing activities to fit the tablet format, and I was able to spend a little less time. It still took up to a couple of hours to prepare an in-class activity for which I had, in previous years, relied on photocopies or the textbook, but most days I was able to prepare the tablet activity in about an hour. It took some time for me to decide how the student’s tablet display should complement the overhead display. I’m not sure I have that fully mastered yet.

3. “How do you think the students adapted to using the technology in the classroom? It seemed to me at some point that the technology was not really in the students’ hands, but in yours. That is, that the students just got used to arriving to class (and once the bugs were worked out) students took 30 seconds to login, calibrate and get to work. In that way, it seemed that you were the user of technology, and the students were just doing what they would normally do on paper.”

Most students did just as you said, but we always seemed able to find a new bug, or something that didn’t work as expected.

4. “Do you think you were able to provide more 1:1 with students? Do you think you were able to tell when students were struggling, bored, ahead, off task, etc.. How would you compare your ability to do so before coming to Siebel? I personally think you were extremely cognizant of your students before the technology, which only (in my view) makes it more difficult to assess its ability to enhance your overall “knowledge of the classroom.” ”

One real advantage of the dashboard (when I would force myself to take advantage of it) is that students could work together, and I could see what they were doing, but they didn’t feel me breathing down their necks.

Having the teacher standing close by can cause two equally undesirable consequences: some student might get too dependent on feedback from the teacher, and not learn on their own or work with partners; other students might clam up for fear of being caught making a mistake. By maintaining a distance, I forced students to be independent and work with classmates, without intimidating them. By watching their work, however, I was able to judge when an intervention might be necessary.

5. “What do you think the students’ perception of using the technology was?”

Some liked it, some didn’t. You saw their survey results, so you probably have a more accurate impression than I do. In my experience, no matter what I try, some are going to appreciate it, and some are going to roll their eyes and try to survive until the ordeal is over. I find it humorous that students who dress radically, act radically, and challenge authority and convention at every turn are EXTREMELY conservative in their expectations for the classroom: DON’T EVER CHANGE ANYTHING.

6. “One of your classes, the 2PM class, was the class that had the most experience/usage of the system, both fall 2007 and spring 2008. Did they do any better in terms of grades in the course? Were there any noticeable differences between classes or even classmates that you think the dashboard or even the (snapshot of students final states of a worksheet as we once printed out) made a difference at all? The 12PM class was the only class I was able to poll BEFORE (their first day in Siebel, and AFTER, the conclusion of the Spring semester) I wonder what differences you saw, if any, in these classes?”

This is a tough one. The 2:00 class was more lively, and had a few highly talented and highly motivated students, and about an equal number at the other end of the talent/motivation scale. The 12:15 class had probably more talent and motivation, on average, than the 2:00 class. Judging the impact of a pedagogy change on these students is very difficult. I can say that there were no marked changes in grades from one semester to the next, either for the group that used tablets for most of both semesters or for the group that used tablets in the second semester only.

B.4 Manish Agrawal, Teaching Assistant for CS125TPC

My Experience with Teaching the Class with a Tablet PC

When I started using a Tablet PC for teaching, it was a bit clumsy and nonintuitive, but slowly I got used to the software and the hardware interface. I used to have my basic lectures in a PDF format that could be uploaded to into the Tablet software and I used to annotate on top of the basic slides while lecturing. The format of the basic slide could be varied based on the content I was teaching. There were also instances when I used the tablet as a white board. Even then, one of the major advantages that it provided was *automatic archiving*. After the lecture, I had class notes that I just cleaned for 10 minutes and then distributed among students. One of the drawbacks of this kind of technology is the learning curve for the instructor and the expectation of legible handwriting.

All the students in the class also had tablets, and they had the same basic slides on top of which I was annotating. I could give exercises, and there was a *dashboard* at the instructor's desk on which I could see their progress in *real time*. At the time I was teaching, the dashboard was under development and it only showed the exact images of my students' desktops. It seemed like a *cool idea* but at the same time I had an information overload. I had a big screen with 20 small windows, and it required a lot of concentration to do something useful with it. Sometimes I was more productive if I just walked through the class and saw what the students were doing. It would have been more helpful if I could also see some derived or processed information of the student activity on the dashboard.

Overall, it felt like a cool technology that could take off. I guess one of the biggest hurdles for me as an instructor and for the students in the class was to get used to the new interface. Also, I felt an instructor had to do more preparation before the lecture to make complete use of the technology. It had a lot of features, but they could only be harnessed with the right kind of content, quizzes and exercises. In a traditional blackboard or whiteboard teaching, a teacher could just flow from one technique to the other, because the infrastructure offers minimal support. Here, all the extra support, features and tools required special lecture plans to actually be useful. I would conclude by saying that researches in educational technology should still continue to explore this direction as it does show potential for utility. It may be more useful for high schools and beginning college classes since the course content is usually static and the student population is large. The experts in this kind of technology may also develop some basic generic content and exercises that may be used by the instructors as a base line for their own lecture contents.

Appendix C

e-Fuzion Documents

C.1 Summer 2002 Documentation

The e-Fuzion documentation for the summer 2002 pilot study was written by the teaching assistant assigned to the course, Zackary Runner, and edited by Chad Peiper for inclusion into this dissertation.



Student Edition Software Documentation

June 16, 2002

C.1.1 Contents

1. Installation
 - (a) Windows 9x/NT/2000/XP
 - (b) Linux, Unix, Mac OSX
 - (c) Web-based usage
2. Using e-Fuzion
 - (a) Taking notes
 - (b) Connecting to the server
 - (c) Following the instructor
 - (d) Posting questions and answers
 - (e) Answering class polls
3. Troubleshooting
 - (a) FAQ
 - (b) Support

C.1.2 Installation

Windows 9x/NT/200/XP

The instructor edition of e-Fuzion is designed to run on any Microsoft operating system with the .NET Framework installed. Currently, the .NET framework is available for the following versions of Windows:

- Microsoft Windows® 98
- Microsoft Windows NT® 4.0 (SP 6a required)
- Microsoft Windows Millennium Edition (Windows Me)
- Microsoft Windows 2000 (SP2 Recommended)
- Microsoft Windows XP Professional
- Microsoft Windows XP Home Edition

Take special note of the patches required/recommended for Windows NT and Windows 2000. The .NET Framework redistributable can be obtained from the Microsoft website (<http://www.microsoft.com>).

Once the .NET Framework is installed, install e-Fuzion by double clicking on the setup icon and following the on-screen instructions. Run e-Fuzion by either double clicking on the desktop icon or through Start → Programs → e-Fuzion → e-Fuzion SE. Be sure to check our website for the latest updates.

C.1.3 Linux, Unix, Mac OSX

Both Linux and Unix will be able to use the client edition of e-Fuzion through Microsoft Terminal Services. However, Microsoft does not provide client software for these platforms. The user will have to obtain a third-party client to connect to the Terminal Services Server. Mac OS X users will be able to connect to the Terminal Services server via Internet Explorer.

More information will be posted soon with regard to obtaining third-party software and connecting through the web.

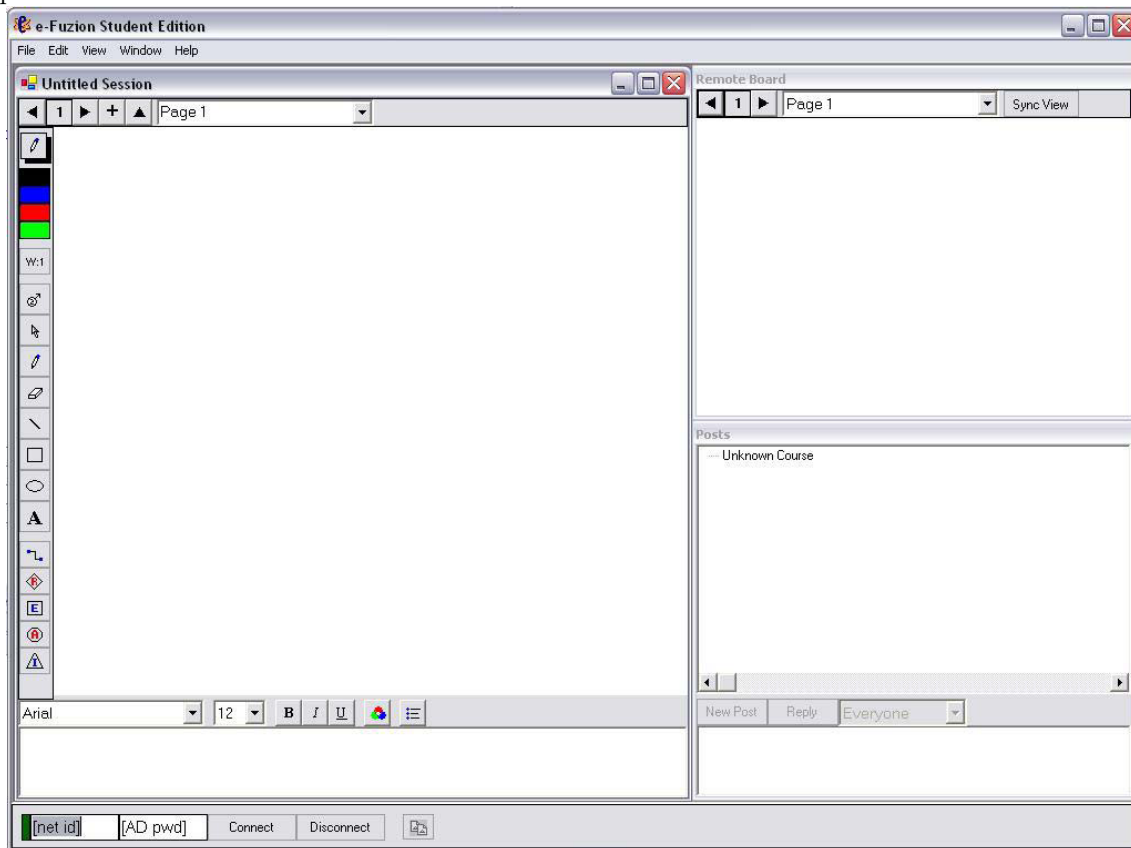
Currently, Linux, Unix, and Mac OSX users are unable to natively run e-Fuzion. However, the incompatibility should be mediated as soon as the .NET framework is implemented for these platforms. These projects are currently being undertaken by Mono (<http://www.go-mono.com>) and Corel (<http://www.corel.com>).

C.1.4 Using e-Fuzion

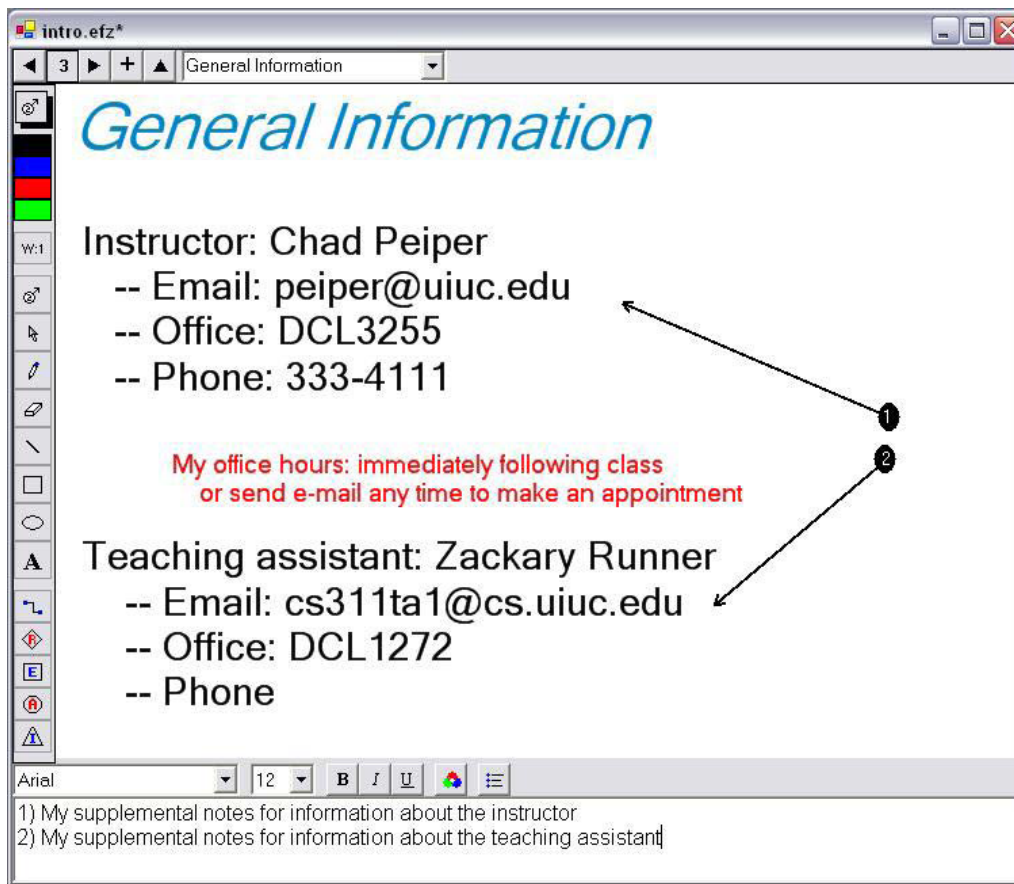
Taking notes

The e-Fuzion system provides a rich set of tools that are customized for each specific course. Each tool is designed to minimize the amount of time spent taking notes so more time can be spent learning what you

are writing. These are the same tools that are provided to the instructor for creating and delivering the lecture. The interface is designed after industry-standard vector drawing programs to make it as intuitive as possible.



Each page of drawn notes is supplemented by textual notes as well. The pointer tool may be used to correlate the two together. The index of the pointer is automatically incremented each time a new pointer is created. You can customize your color palette by double-clicking on the colors.



Additionally, any part of your notes may be cut, copied, or pasted onto other pages by using the select tool.

Connecting to the server

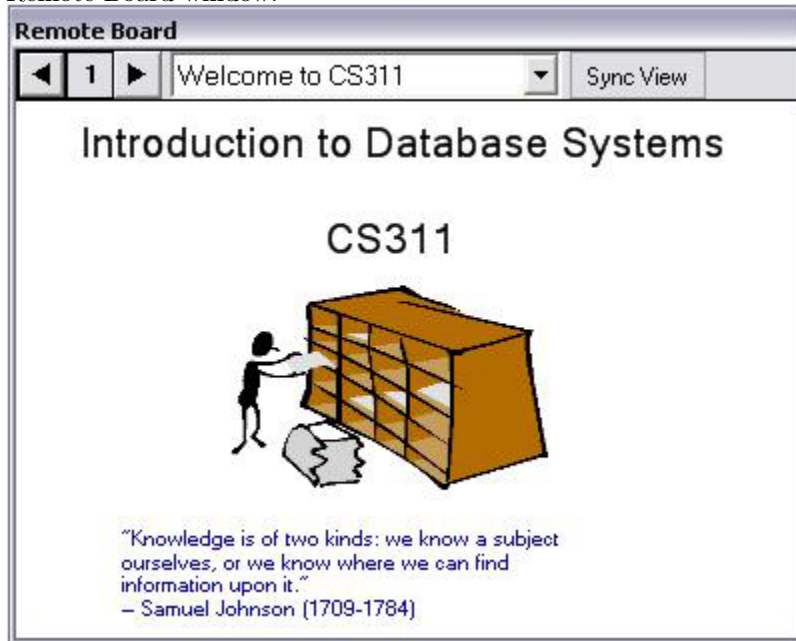
Connect to the e-Fuzion server by entering your UIUC Active Directory login/password and clicking on the Connect button. Keep in mind that you must be connected to the Internet to successfully access the server. However, you may open, edit, and save any of your notes without connecting. You will need to connect to copy the instructor lecture slides, post questions, and participate in class polls.

Linux, Unix, and Mac OSX users

Before connecting to the e-Fuzion server, you will need to log on via Terminal Services. After Terminal Services is properly installed on your system, connect to efuzion.cs.uiuc.edu. Once presented with a logon screen for Windows 2000 Terminal Services, login with your UIUC Active Directory login/password. Once logged in, simply follow the instructions above as if you were using a Windows PC

Following the instructor

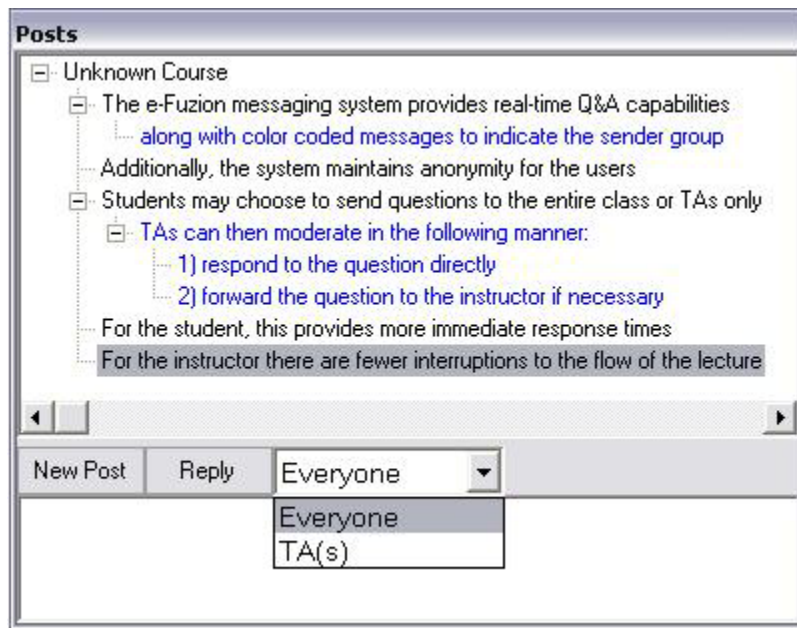
Once you are connected to the server, everything the instructor draws on the board will be reflected in the Remote Board window.



As the instructor switches pages, your view will follow along unless you begin manually browsing through the pages yourself. You will have access to every previous slide that the instructor has covered. To resynchronize your page with the instructors, just hit the Sync button. You may also copy the instructors slides to your own notes by selecting what you want copied and clicking on the copy button at the bottom of the screen. If nothing is selected in the Remote Board, everything on the page will be copied.

Posting questions and answers

The message posting system in e-Fuzion is very similar to the class newsgroups you are already familiar with.



When posting a message, you have the choice of either sending the question to everyone in the class or only to teaching assistants. By highlighting the most recent post, your window will automatically highlight future posts. Posts are also color-coded:

- Students: BLACK
- TAs: BLUE
- Instructor: RED

Answering class polls

On occasion, the instructor may choose to send a poll out to the class. When this happens, you will see the following window appear on your screen:

Quiz Question

Question:

This is an example of a poll question involving the Entity-Relationship model in database systems: Which of the following is correctly labeled?

Text Answer Graphical Answer Send!

☒ a ☐ b ☐ c ☐ d ☐ e

ER Model Basics Each attribute has a ~~ad~~omain
Each entity set has a ~~de~~key

ER Model Basics Each attribute has a ~~ad~~omain
Each entity set has a ~~de~~key

C.1.5 Troubleshooting

FAQ

Q. Why doesn't it work when I try to install?

A. The e-Fuzion system requires the Microsoft .NET Framework to be installed first. Instructions for doing so can be found in the Installation chapter.

Q. How do I find out my Active Directory login/password?

A. Your Active Directory password is the same one you use to login to any Windows system on campus. Visit the Active Directory link at <http://passwords.ad.uiuc.edu> to change or create your password. Use your NetID as your login.

Q. Why can't I connect to the server?

A. You must first establish an internet connection through wireless networking or otherwise. Make sure you are entering your UIUC Active Directory login/password (Note: your Kerberos and Bluestem passwords will not authenticate you). When you are not in class, the e-Fuzion server may not always be accessible.

Q. Why do I get disconnected from the server immediately after I connect?

A. Make sure you have the latest version of the e-Fuzion student edition. Check for updates at efuzion.cs.uiuc.edu.

Q. I am connected to the server. Why cant I see what the instructor is writing or post messages?

A. You may have unexpectedly lost your connection. Try clicking Disconnect and then clicking Connect to reestablish your connection.

C.2 e-Fuzion Vector Graphics Library Write Up

The Vector graphics library and documentation below was created and written by Patrick Bristow and edited by Chad Peiper for inclusion into this dissertation. Patrick Bristow was one of the two original authors of the e-Fuzion software.

C.2.1 Introduction

When e-Fuzion first began, we were tired of seeing a rasterized view of the world. Every existing shared whiteboard software sent to the clients bland raster images of what the person at the "server" was doing. For the purposes of a teaching tool, this was ineffective; what we needed was interaction. We endeavored to produce a piece of software that would allow the server and client to interact with every object within the drawing or document. In reading this particular document, I hope that you'll feel like you're getting to interact with the inner-workings of the vector library. While the source code is not openly available, I hope that enough snippets and descriptions will be included to give you a firm understanding of what it takes to develop effective uses of the library.

C.2.2 Why not use the Dia® or PowerPoint® libraries?

Simply put, neither Dia nor PowerPoint offers the features necessary for real-time interaction. The design of the e-Fuzion Vector Library has always revolved around allowing graphics to be sent and updated quickly, without having a noticeable effect on either the client or server. Each graphic's feature set has to be full, but the data required to reproduce the graphic has to be minimal. Pages and graphics have to be unique on both ends of the network both the client and server can interact with the graphics without ruining the other's work. To our knowledge, there was nothing that effectively met our needs; in the end, we did our best to learn from others' mistakes and create the perfect vector library for our needs.

C.2.3 It takes all system types

Following all the classes, structs, and enums that make up the VectorDrawing namespace can give one a great headache. Thus, I'm going to start from the bottom up, from VGraphic to VectorSketcher.

- VGraphic: Every graphic or stroke of a pen that ends up on the screen is derived from this abstract base class. It provides almost all of the functionality needed for a simple graphic except for the Display method and constructor(s).
- GraphicNode: Each graphic is contained in a GraphicNode. At current, the node is a superfluous object who's destiny is to contain the created and deleted times for every graphic, allowing full, movie-like replay possible.
- GraphicNodeList: A collection with a small set of features that assist in managing all of the GraphicNodes a page might contain.
- Page: Just what it sounds like, a Page is simply a page in a workbook. It contains a GraphicNodeList and an Ink object, and provides methods to assist in dealing with moving graphics from one network location to another.
- PageSet: A collection to help with the confusing nature of a Page's ID vs. its number. Human interaction dictates that every Page must have a numerically ordered number, but for network safety, every page has a unique ID. PageSet provides high-speed lookup between the two.
- VGInterface: When a graphic resides in memory on a certain page, it is considered to have a "culture" - the page on which it resides. VGInterface provides an interface for complex graphics to exist properly within its culture.
- VectorSketcher: This top-level class does the majority of the grunt work, such as deciding when to select graphics, cut/copy/paste, enabling the ink overlay, and managing graphics.
- SketcherHelper: As the name might imply, SketcherHelper provides assistance for network syncing between two sketchers at opposite ends of the Earth.
- Aside from all of the above, there are seven other classes, structs, and enums in VectorDrawing. Each of these has sufficient explanation included in the XML comments. Many of them deal with the individual data packets send over the network.

C.2.4 VGraphic, simplicity through complexity

VGraphic is the abstract base class from which all graphics must be derived. Its design makes it distinctly unique from any other graphics classes because of the goals it is designed to achieve. First, the bandwidth consumed to move the graphic from computer to computer must be kept at a minimum. Second, the functionality of the base class must allow complex, interactive graphics for effective instructional and communication tools (e.g. Rick Ink). Finally, the graphic must have the appropriate methods and overall implementation to allow movie-like playback of a session. This last criteria focuses on features to be implemented in the near future, but nonetheless, it is an important design goal.

The subtitle for this section, "Simplicity though complexity", is a testament to how much effort it takes to create a basic graphic. Take, for instance, creating an 'ellipse' graphic; to do so, two methods must be

implemented: a constructor and the ‘Display’ method. In the VRectangle class, the two methods form a total of 17 lines. The reason is because all of the methods and properties of VGraphic are already implemented for a typical symmetric graphic (e.g. An ellipse or rectangle).

There are two constructors to most VGraphics, each with implicit meaning. The first takes a variety of arguments, including pen width, foreground color, and the VGInterface. Calling this constructor implies the graphic is being manipulated, and its current state is set to “Moving Point 2.” This means that Point 1 is fixed and Point 2 is being dragged by the mouse. This constructor is designed to be called after the mouse has moved some distance (to defeat one pixel by one pixel graphics), but this can be side-stepped by setting the P1 and P2 arguments the same. The second constructor takes the VGInterface as its only arguments, and sets its current state to “None.” This constructor doesn’t have any implicit meaning and is sometimes used during custom de-serialization.

After calling the constructor, the MouseMove and MouseUp methods allow quick calling from delegates of OnMouseMove and OnMouseUp. MouseUp must be called in order to reset the current state of the graphic to “None.” For most graphics, the PointF argument to both methods correlates to Point 2 of the graphics; in the case of symmetric graphics, Point 2 is the bottom right and Point 1 is the top left.

MouseDown serves two purposes: first, it checks the location of the provided point to detect if it is inside the graphic or on part of the selection interface. Second, it sets the internal flags as necessary if the graphic has “been hit” and returns true. Thus, each time the mouse is clicked, MouseDown can be called on each graphic on the screen until one returns true.

A handful of other methods and properties enable outside manipulation. The “IsChild” property reflects whether the graphic has been marked as the child of another graphic; this affects whether the graphic can be selected or performs its own drawing. A graphic which is marked as a child is virtually transparent to the VectorSketcher class except during copy/paste (a recursive-like operation). Finally, a small set of static utility methods are available to save time and improve performance.

C.2.5 VGInterface: it doesn’t start with ‘I’

The Vector Graphics Interface provides a set of methods and data to allow a graphic to be aware of its “culture”—the page in which it resides and the sketcher which it is slave to. The class is not an interface in the strict programming sense of the word. However, it allows a VGraphic to have a single reference to all of its culture. Once the culture has been applied to the graphic, it is “aware” of its surroundings and can perform functions. Functions can include (but are not limited to) rendering itself, modifying child graphics, creating or deleting new graphics on the same page, and even deleting itself from the page.

The VGInterface (or VGI) is applied via a VGI property in the VGraphic. The ‘set’ part of this property allows the graphic to become aware of when it has entered a new culture. At times, it is necessary for the graphic to be aware of its new culture and change some of its properties. For instance, after de-serializing a VStroke (a wrapper for a Rich Ink Stroke object), the stroke must be added back to the Ink object for that page; VGI allows that to happen.

C.2.6 Conclusion

VectorSketcher and SketcherHelper are also extremely important classes. However, they are well documented with XML-style commenting and should not be as complicated to understand as the above topics. Also, those two classes provide most of the interfacing to software that would use the Drawing Library; thus, they are major construction lately to provide a cleaner, neater interface. In time, those will be documented as well.

Appendix D

Whiteboard Lecture File Examples

D.1 Cinda Heeren – CS225

The first file is an example of a post lecture document with instructor annotations. The second document sample is a comparison of pages derived from the original skeletal version of the presentation followed by the same page with instructor annotations.

Announcements:

•Midterm 2 will be returned in section this week.

•mp6 due tue, 11/27, 11:59p (midnightish).

•mp7 available within 24 hours... start early.

•final exam time has still not been announced.

Today:

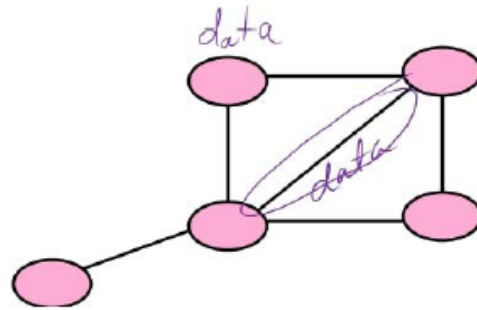
Graphs

Theory

Algorithms

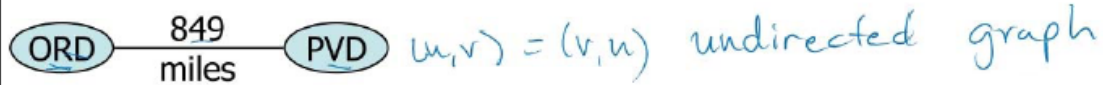
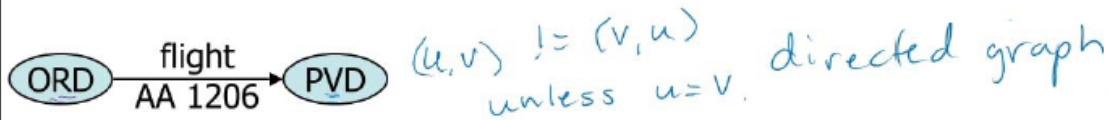
Implementation

Graphs: Mathematically represented as a pair of sets (V, E) , where V is the collection of "vertices" and E is the collection of "edges."



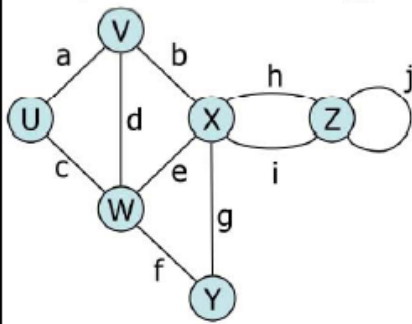
Applications: http://www.aisee.com/graph_of_the_month/archive.htm

Graphs: directed and undirected edges $E = \{(u, v) : u \in V, v \in V\}$



notice :
undirected graph can be modelled as
a directed graph.

Graphs: more terminology



endpoints of edge b are vertices V, X.

edges abd are incident on vertex v

the degree of vertex x is 5

x, y are adjacent vertices

h, i are multi-edges

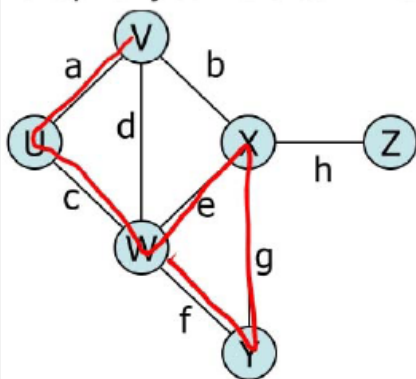
j is a self-loop

x, z; u, w ...
u not adjacent to x.

these do not exist in "simple" graphs

our graphs are by default "simple."

Graphs: yet more terminology



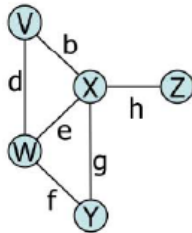
A path is an alternating sequence of edges and vertices that starts and ends at vertices.

$V, u, c, w, e, x, g, y, f, w$
 $(V, u)(u, w)(w, x)(x, y)(y, w)$

A simple path is a path in which each vertex appears at most once.

V u w x

Graphs: Toward implementation...(ADT)



Functions: (merely a smattering...)

insertVertex(pair keyData)
insertEdge(vertex v1, vertex v2, pair keyData)

removeEdge(edge e);
removeVertex(vertex v);

Data:

Vertices

Edges

+ some structure that reflects the connectivity of the graph

numVertices
numEdges } convenience

incidentEdges(vertex v); → iterator over incident edges.

areAdjacent(vertex v1, vertex v2); → probably a one liner edge check

origin(edge e);

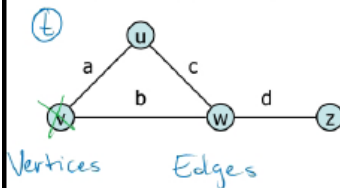
destination(edge e);

} directed graphs



$$|V| = n \quad |E| = m$$

Graphs: Edge List (a first implementation)

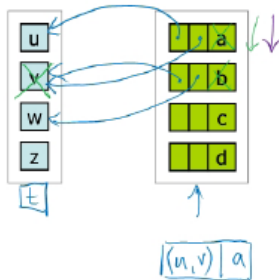


Vertices

Edges

Some functions we'll compare:

- insertVertex(vertex v) $O(1)$ assuming a reasonable vertex structure.
- removeVertex(vertex v) $O(m)$
- areAdjacent(vertex v, vertex u) $O(m)$



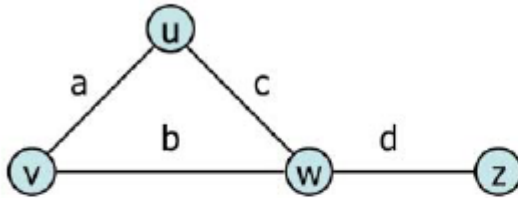
aside: how bad is $O(m)$? (in terms of n)

$$m \leq \frac{n(n-1)}{2} = O(n^2)$$



(a second implementation)

Graphs: Adjacency List

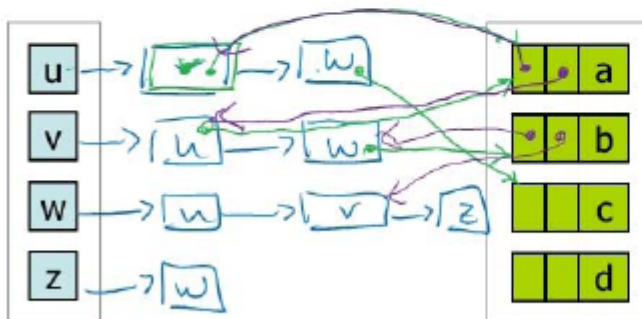


Some functions we'll compare:

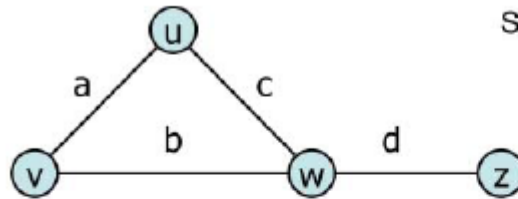
`insertVertex(vertex v)`

`removeVertex(vertex v)`

`areAdjacent(vertex v, vertex u)`



Graphs: Adjacency Matrix



Some functions we'll compare:

`insertVertex(vertex v)`

`removeVertex(vertex v)`

`areAdjacent(vertex v, vertex u)`

u	
v	
w	
z	

		a
		b
		c
		d

Announcements:

- Midterm 2 will be returned in section this week.
- mp6 due tue, 11/27, 11:59p. (late handin still available for 20% penalty)
- mp7 available start today.
- final exam 12/14, 1:30-4:30p.

Today:

Graphs - representation and traversal

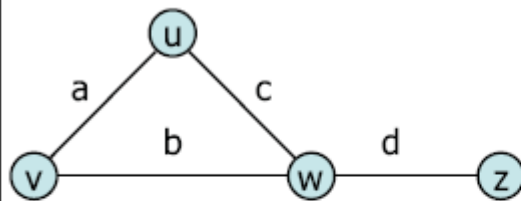
Announcements:

- Midterm 2 will be returned in section this week.
- mp6 due tue, 11/27, 11:59p. (late handin still available for 20% penalty)
- mp7 available start today.
- final exam 12/14, 1:30-4:30p.
- Code challenge 11/29 7p see course website then.

Today:

Graphs - representation and traversal

Graphs: Adjacency List

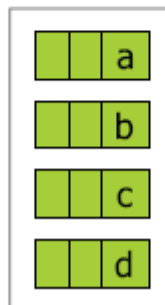


Some functions we'll compare:

`insertVertex(vertex v)`

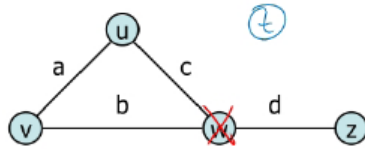
`removeVertex(vertex v)`

`areAdjacent(vertex v, vertex u)`



Graphs: Adjacency List

$$|V| = n \quad |E| = m$$



Vertex

Edge



E

Keep
degree of
vertices

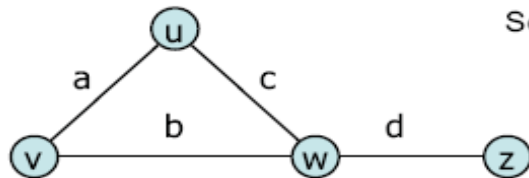
Some functions we'll compare:

insertVertex(vertex v) $O(1)$

removeVertex(vertex v) $O(\deg(v)) = O(n)$

areAdjacent(vertex v, vertex u) $O(\min\{\deg(u), \deg(v)\})$

Graphs: Adjacency Matrix

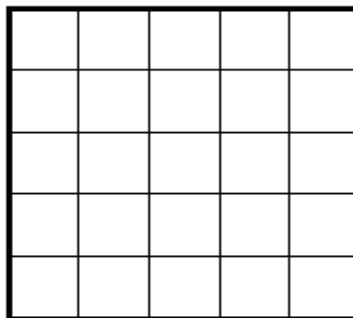
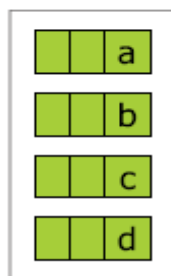
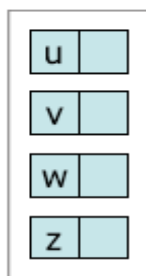


Some functions we'll compare:

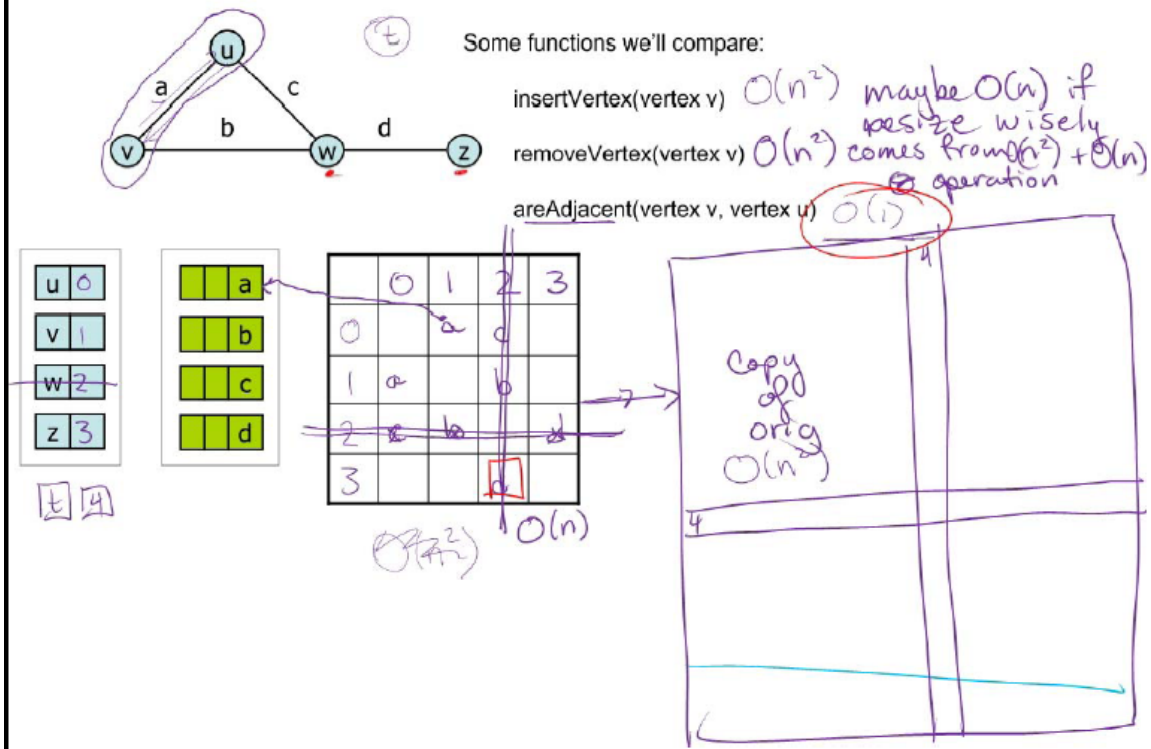
insertVertex(vertex v)

removeVertex(vertex v)

areAdjacent(vertex v, vertex u)



Graphs: Adjacency Matrix



Graphs: Asymptotic Performance

<ul style="list-style-type: none"> n vertices, m edges no parallel edges no self-loops Bounds are big-O 	Edge List	Adjacency List	Adjacency Matrix
Space	$n + m$	$n + m$	n^2
incidentEdges(v)	m	$\deg(v)$	n
areAdjacent(v, w)	m	$\min(\deg(v), \deg(w))$	1
insertVertex(o)	1	1	n^2
insertEdge(v, w, o)	1	1	1
removeVertex(v)	m	$\deg(v)$	n^2
removeEdge(e)	1	1	1

$$|V| = n \quad |E| = m$$

$$m = O(n^2)$$



Graphs: Asymptotic Performance

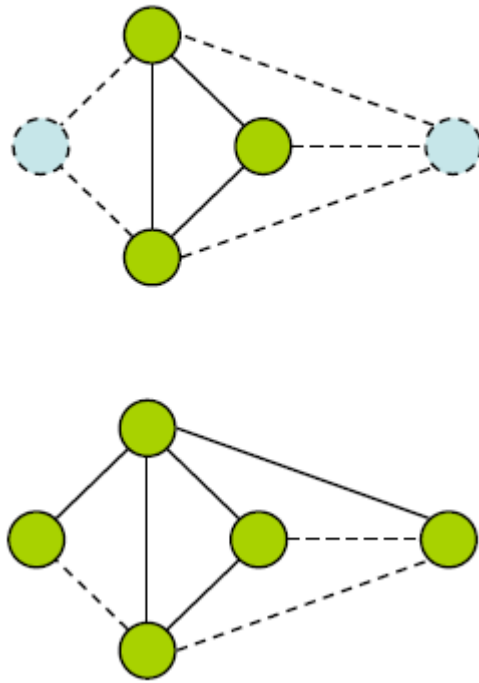
preference for one implementation over another

depends on density of graph (# of edges relative to # of vertices)

• n vertices, m edges • no parallel edges • no self-loops • Bounds are big-O	Edge List	Adjacency List	Adjacency Matrix
Space	$n + m$	$n + m$	n^2
incidentEdges(v)	m	$\deg(v)$	n
areAdjacent(v, w)	m	$\min(\deg(v), \deg(w))$	1
insertVertex(o)	1	1	n^2 n
insertEdge(v, w, o)	1	1	1
removeVertex(v)	m	$\deg(v)$ <small>no worse than n</small>	n^2 n
removeEdge(e)	1	1	1

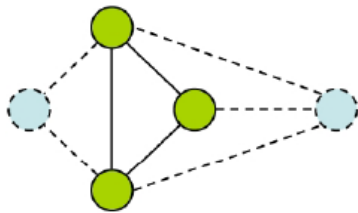
→ traverse edge list

Graphs: Subgraphs

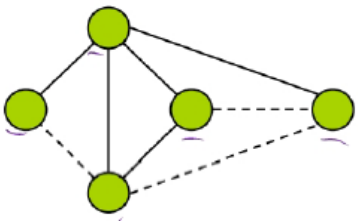


Graphs: Subgraphs

$G=(V,E)$

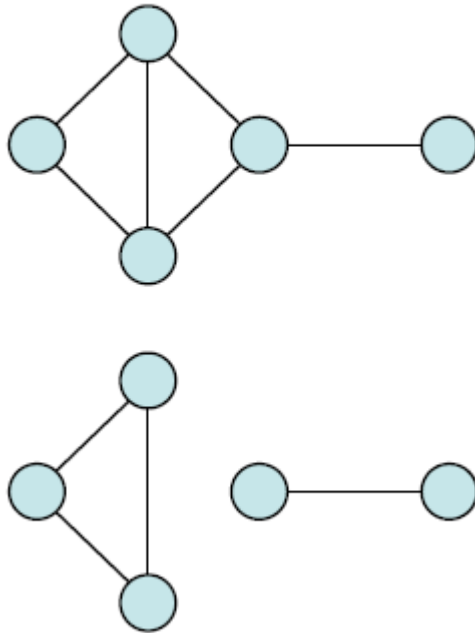


a subgraph of G is a graph $G'=(V',E')$ where
 $E' \subseteq E$, $V' \subseteq V$ and
 $(u,v) \in E' \longrightarrow u \in V' \wedge v \in V'$

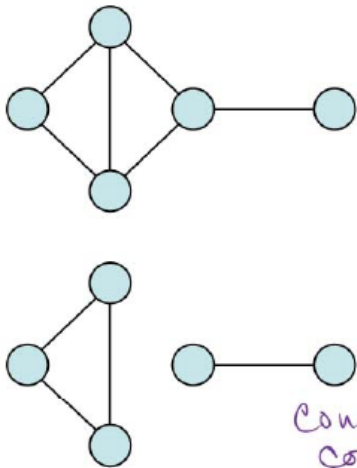


a spanning subgraph of G is a
 subgraph $G'=(V',E')$
 where $V'=V$.

Graphs: Connectivity



Graphs: Connectivity



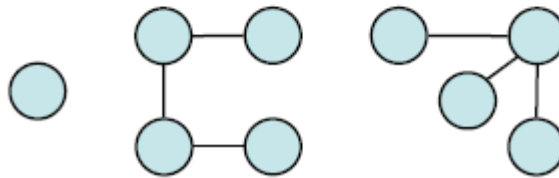
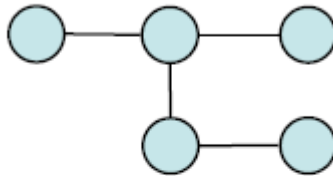
a "connected graph" is a graph so that if $u \in V$ and $v \in V$, a path exists between u + v .

not connected

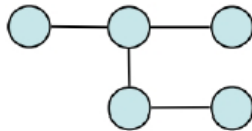
2 connected components.

connected component: maximally connected subgraph. (can't add more vertices + maintain connectivity)

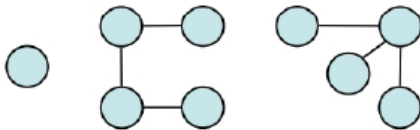
Graphs: Trees and Forests



Graphs: Trees and Forests

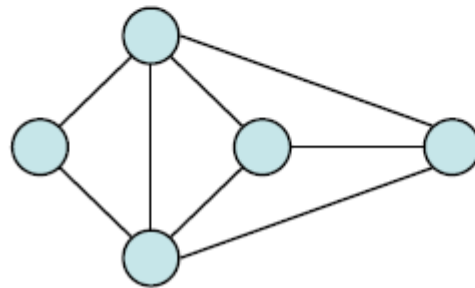
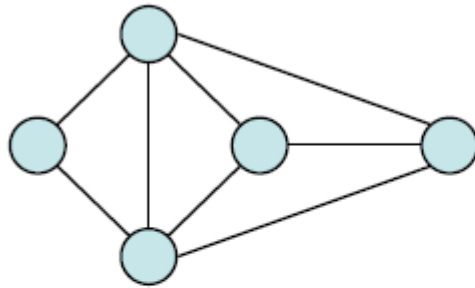


a connected acyclic graph is
a tree.



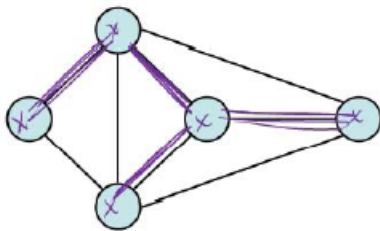
a forest is an acyclic graph.

Graphs: Spanning Trees and Forests

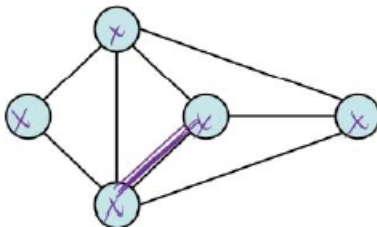


Graphs: Spanning Trees and Forests

$G(V, E)$

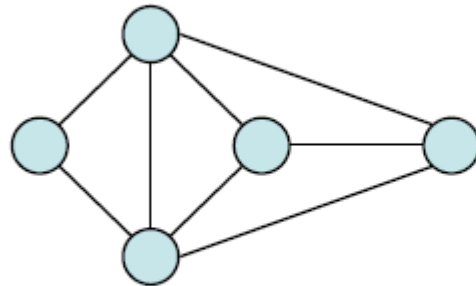


Spanning tree $G' = (V', E')$
connected acyclic subgraph
with $V' = V$.



acyclic subgraph $V' = V$

Graphs: Traversal - DFS



<http://www.cs.duke.edu/csed/jawaa2/examples/DFS.html>

<http://www.student.seas.gwu.edu/~idsv/idsv.html>

Algorithm DFS(G)

Input: graph G

Output: labeling of the edges of G as discovery edges and back edges

```
For all u in G.vertices()
    setLabel(u, UNEXPLORED)
For all e in G.edges()
    setLabel(e, UNEXPLORED)
For all v in G.vertices()
    if getLabel(v) = UNEXPLORED
        DFS(G,v)
```

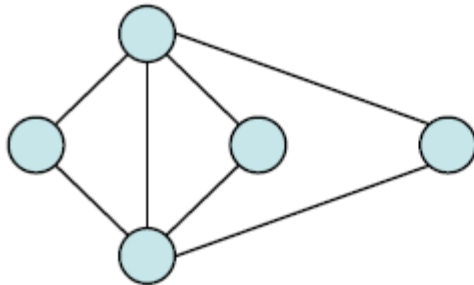
Algorithm DFS(G,v)

Input: graph G and start vertex v

Output: labeling of the edges of G in the connected component of v as discovery edges and back edges

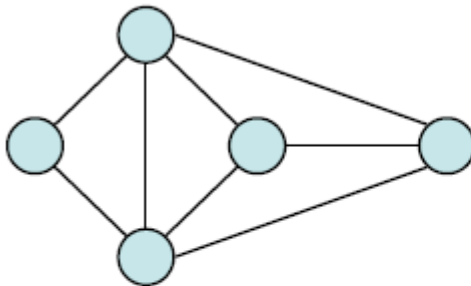
```
setLabel(v, VISITED)
For all e in G.incidentEdges(v)
    if getLabel(e) = UNEXPLORED
        w = opposite(v,e)
        if getLabel(w) = UNEXPLORED
            setLabel(e,DISCOVERY)
            DFS(G,w)
        else
            setLabel(e,BACK)
```


Graphs: DFS example



A	B	C	D	E
B	A	C		
C	B	A	D	E
D	A	C		
E	A	C		

Graphs: DFS Analysis



setting/getting labels

every vertex labeled twice

every edge is labeled twice

querying vertices

total incident edges

TOTAL RUNNING TIME:

Solving Equations Using Logs

- **Exponential Equations:** Make sure you use a common base; combine all exponential expressions in one term before taking logarithms.

» Example:

$$2^x + 3 = 2^{x+3} \rightarrow$$

- **Log equations:** Use properties of logarithms to re-write complicated expressions, then use exponentiation to solve.

» Example:

$$2 \log(3x) - \log(x) = 4 \rightarrow$$

Exploring Functions and Solving Equations!

Solve each equation below (to the nearest 0.01). Be sure to check your answers!

1. $5x^3 + 2 = 10$

7. $11^{6x-3} = 80$

2. $5 \cdot 3^x + 2 = 10$

8. $2^x = 0.0053$

3. $6^x = 3$ vs. $x^6 = 3$

9. $5000(1 + 0.015)^{2x} = 6000$

4. $196^{x^2+1} = 537,824$

10. $\sqrt{x+1} - \sqrt{x} = \frac{1}{2}$

5. $3(10^x) = 2(5^x)$

11. $\sqrt{x+2} - x = 2$

6. $\log_2(x+1) - \log_2 x = 2$

12. $x+8 - x = 2$

Graph $f(x) = 3e^{2x}$ and $g(x) = 3e^{2(x+2)}$ (calculator graphs OK; you don't need to sketch here). Discuss how the graphs differ. How do the graphs of $f(x)$ and $h(x) = 5e^{2x}$ differ? **THINK GEOMETRY...**

Graph the *power function* $f(x) = x^{0.5}$ and the *logarithm function* $g(x) = \log x$. Discuss as many similarities and differences as you can think of. In your discussions, be sure to touch on increasing/decreasing, concavity, growth rates, and asymptotes.

Repeat the exercise, using this time the *power function* $f(x) = x^{-1/2}$ and the *exponential function* $g(x) = (1/2)^x$.

5 Exploring Functions and Solving Equations!

TRANSFORMATIONS:

You may have seen this earlier: Discuss two *different* “transformations” that relate the functions $f(x) = 3e^{2x}$ and $g(x) = 3e^{2(x+2)}$ [HINT: use a property of exponents to re-write $g(x)$ to see the second transformation... the first one should be obvious!]

Multiply-Multiply Property: for these problems, $y = f(x) = ax^p$ for some power p and some real number a . You may assume that $x > 0$.

1. Show that, if $f(x) = ax^p$, then $f(kx) = m \cdot f(x)$ (multiplying x by something causes $f(x)$ to be multiplied by something—**find** the “something” that will be in terms of k and p).
2. “Doubling x causes y to triple.” Use this statement to find p (there’s not enough information to find a !).
3. If $f(4) = 1$, and when x doubles $f(x)$ is multiplied by 0.9, find $f(x)$

Note the similarity of problems 1-5!

Solve each equation below (to the nearest 0.01). Be sure to check your answers!

1. $A^2 + A - 6 = 0$

2. $(x+3) + \sqrt{x+3} - 6 = 0$

3. $e^{2x} + e^x - 6 = 0$

4. $(\log_2 x)^2 + \log_2 x - 6 = 0$

5. $x^{-2/3} + x^{-1/3} - 6 = 0$

6. $3x^2 = 4x^5$

7. $3 \cdot 2^x = 4 \cdot 5^x$

8. $3 \cdot 2^{x+1} = 4 \cdot 5^{2-x}$

9. $\sqrt{x+2.5} - x = 0.5$

10. $\sqrt{\ln x + 2.5} - \ln x = 0.5$

11. $4^x + 9 = 6 \cdot 2^x$

12. Solve for y : $\ln y = 1.2 \ln x + 3$

13. $(\log x)^2 + \log x^2 = 15$ (be clever!)

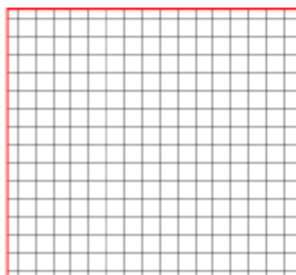
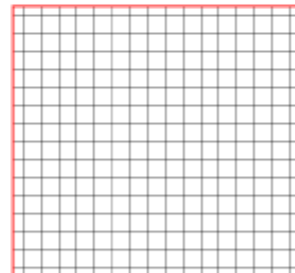
14. $\log_4 x - \log_2 x = 1$

REMINDER BOX: These are "standard" equations for each conic section:

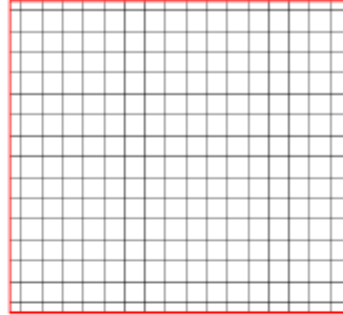
CIRCLE: $x^2 + y^2 = r^2$ HYPERBOLA: $\left(\frac{x}{a}\right)^2 - \left(\frac{y}{b}\right)^2 = 1$ OR $\left(\frac{y}{b}\right)^2 - \left(\frac{x}{a}\right)^2 = 1$

PARABOLA: $4py = x^2$ OR $4px = y^2$ ELLIPSE: $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1$

- What's the equation for a circle with radius 2 centered at $(-3, 6)$?
- How does it compare with a same-sized circle at $(0,0)$?
- What's the equation for a parabola with vertex at $(0, 0)$ and focus at $(1, 0)$?
- How does this help when finding the equation for a parabola with vertex $(-3, 6)$ and focus $(-2, 6)$?
- Give the equation for an ellipse that is 10 units tall, 3 units wide, centered at $(7, -5)$.



- What about a hyperbola with vertices $(1, 4)$ and $(1, 8)$, with foci at $(1, 3)$ and $(1, 9)$?

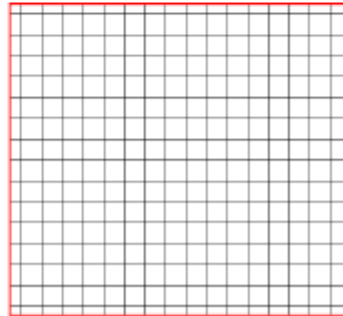


- Consider the equation $\left(\frac{y-3}{4}\right)^2 - (x+2)^2 = 1$. What shape is the graph?

- What hyperbola, centered at the origin, would be congruent to this one?

- Where are the vertices and foci of the hyperbola?

- What is the eccentricity of the hyperbola?



D.3 Spring 2007 Lecture Documents

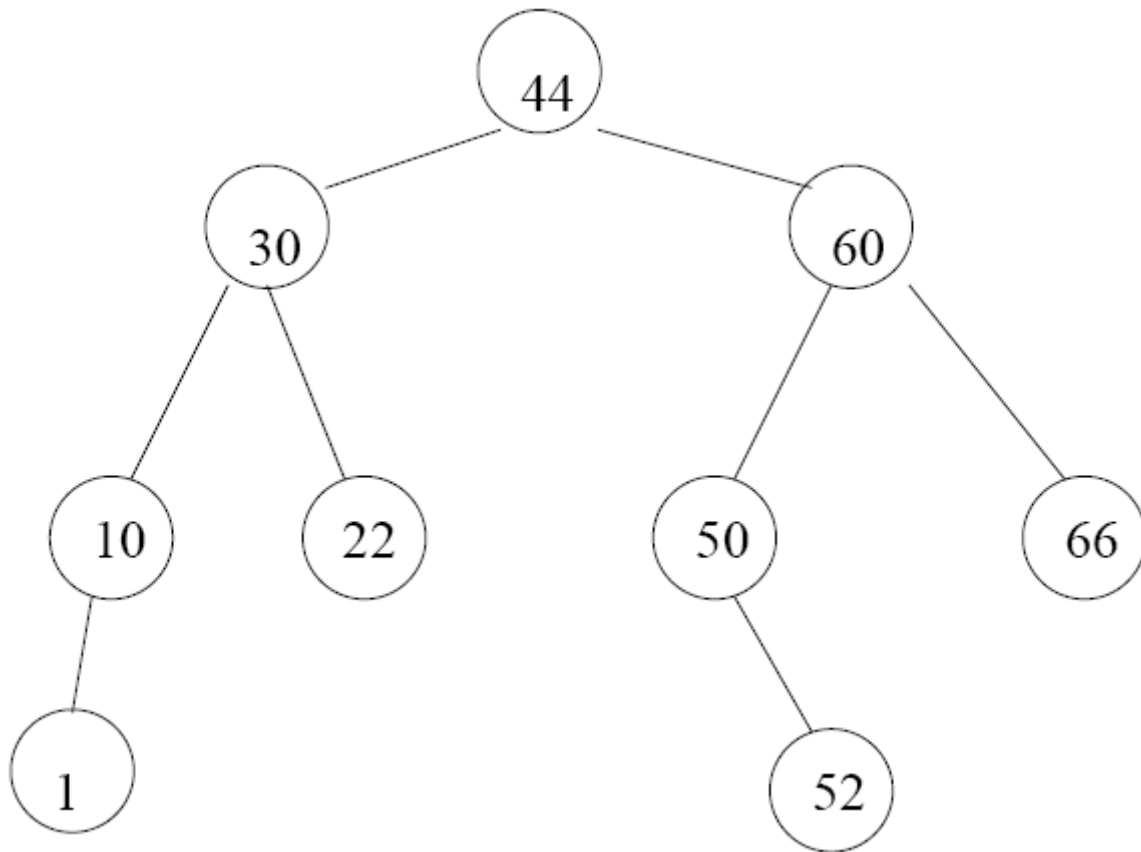
The Spring 2007 pilot study system required an xml file detailing the objectives for the lesson and PDF files which contained assessments for each objective. Each PDF file is required to have the pre-flight exercise for the first page of the document. The next group of pages are the number of assessments * 2 (one page for the assessment, one page for the solution), and a final pair of pages for the objective's post-flight (the post-flight question, and proposed solution). The XML example file from 6.28 is illustrated below:

```
<?xml version="1.0"?>
<objectives>
<objective name="BST" file="bst.pdf"/>
<objective name="Insertion Algorithm" file="bstInsert.pdf"/>
<objective name="Remove Algorithm" file="bstRemove.pdf"/>
</objectives>
```

We have included the first two PDF files for the first two objectives; binary search trees and the BST insertion algorithm.

Pre-flight: Binary Search Tree Question

1) Is the tree below a BST?



2) The worse case Find time for binary search tree is:

- a) $O(1)$
- b) $O(\lg n)$
- c) $O(n)$
- d) $O(n^2)$

BST Tree – Assessment #1 Question

1) In which of the following does a BST always perform better than a normal binary tree:

- a. Insert
- b. Delete
- c. Find
- d. None of the above

2) Given a BST, the IOS is:

- a. The greatest element in the left subtree
- b. The least element in the left subtree
- c. The greatest element in the right subtree
- d. The least element in the right subtree

3) What does a level order traversal do in a BST?

// write your answer below here

BST Tree – Assessment #1 Solution

1) In which of the following does a BST always perform better than a normal binary tree:

- a. Insert
- b. Delete
- c. Find
- d. None of the above

2) Given a BST, the IOS is:

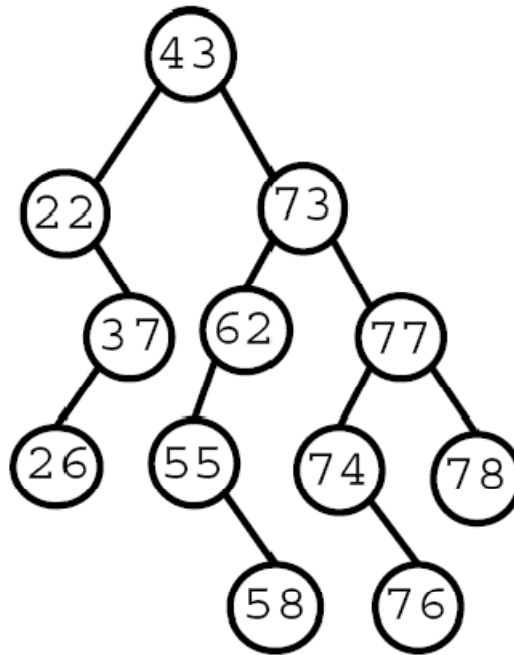
- a. The greatest element in the left subtree
- b. The least element in the left subtree
- c. The greatest element in the right subtree
- d. The least element in the right subtree

3) What does a level order traversal do in a BST?

Visits all elements according to their depth (depth 0, depth 1, depth 2...)

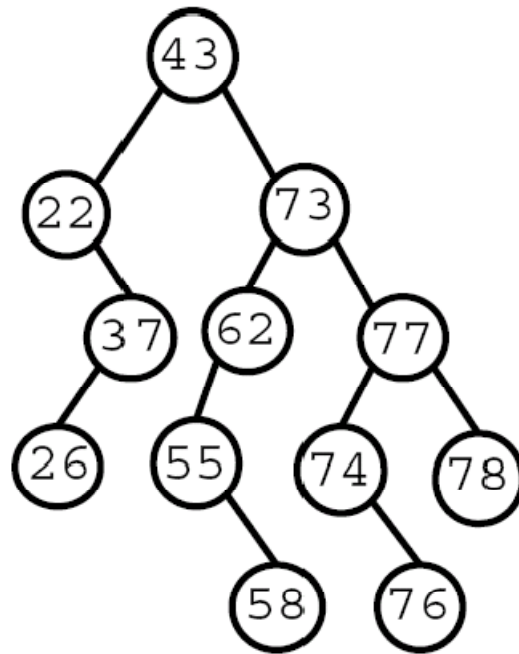
Post-flight: Binary Search Tree Question

Write the list of obtained by taking the post-order traversal of the tree below:



Post-flight: Binary Search Tree Solution

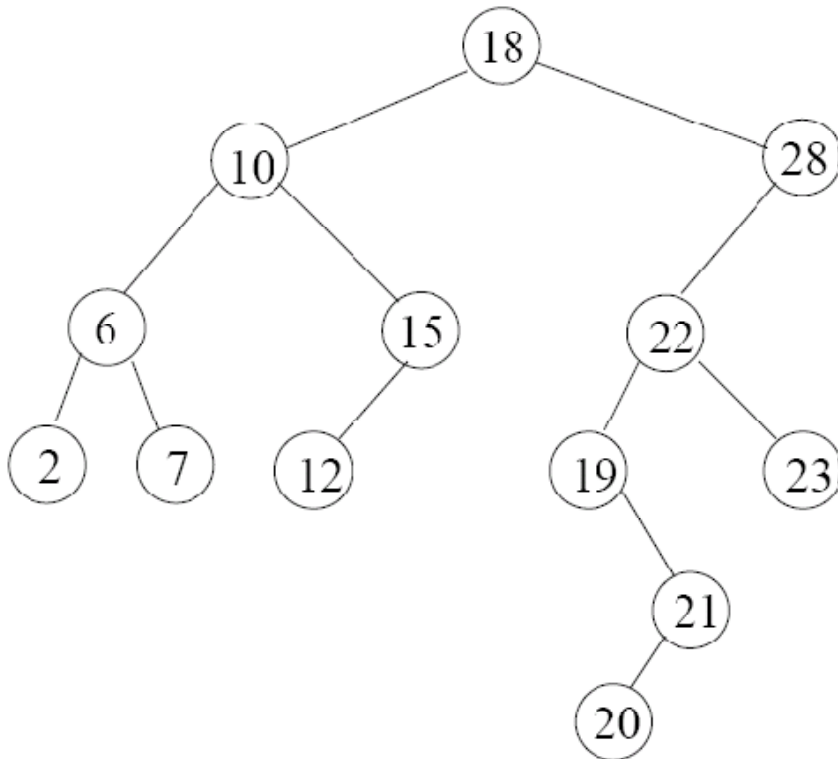
Write the list of obtained by taking the post-order traversal of the tree below:



26, 37, 22, 58, 55, 62, 76, 74, 78, 77, 73, 43

Pre-flight: BST Insertion Question

1) Perform an insertion(17) into the following BST



2) Complete the code for Insertion into a BST:

```
template <class Etype>
void BSTree<Etype>::Insert(const Etype& insElem, TreeNode<Etype>*& TN)
{
    if (TN == NULL)
    {
        TN = new TreeNode<Etype>(insElem);
        size++;
    }
}
```

BST Insertion – Assessment #1 Question

Question 1:

The time taken to insert an element in a Binary Search Tree in the average case is

- a) $O(\log n)$
- b) $O(n)$
- c) $O(n * \log n)$
- d) $O(n^2)$

Question 2:

In a Binary Search Tree with root->element = 10, if we want to insert the element 5, we need to

- a) follow the right pointer
- b) follow the left pointer
- c) replace the root
- d) do none of the above

Question 3 :

The height of a Binary Search Tree after the following sequence of operations is:

```
insert(4)
insert(2)
insert(1)
insert(6)
insert(3)
insert(8)
```

- a) 5
- b) 4
- c) 3
- d) 2

BST Insertion – Assessment #1 Solution

Question 1:

The time taken to insert an element in a Binary Search Tree in the average case is

- a) $O(\log n)$
- b) $O(n)$
- c) $O(n * \log n)$
- d) $O(n^2)$

Question 2:

In a Binary Search Tree with root->element = 10, if we want to insert the element 5, we need to

- a) follow the right pointer
- b) follow the left pointer
- c) replace the root
- d) do none of the above

Question 3 :

The height of a Binary Search Tree after the following sequence of operations is:

```
insert(4)
insert(2)
insert(1)
insert(6)
insert(3)
insert(8)
```

- a) 5
- b) 4
- c) 3
- d) 2

BST Insertion: Post-flight Question

Question: Does the following code work?

```
// Insert
// - parameters : insElem - element to be inserted
//               : TN - a treeNode pointer
// - recursively inserts insElem into tree (does nothing if
//   it is already there
template <class Etype>
void BSTree<Etype>::Insert(const Etype& insElem,TreeNode<Etype>* TN)
{
    if (TN == NULL)
    {
        TN = new TreeNode<Etype>(insElem);
        size++;
    }
    else if (insElem < TN->element)
    {
        recentComp++; totalComp++;
        Insert(insElem, TN->left);
    }
    else if (insElem > TN->element)
    {
        recentComp++; totalComp++;
        Insert(insElem, TN->right);
    }
    // else insElem is in the
}
```


BST Insertion: Post-flight Solution

Question: Does the following code work?

NO!

```
// Insert
//   - parameters : insElem - element to be inserted
//                   : TN - a treeNode pointer
//   - recursively inserts insElem into tree (does nothing if
//       it is already there
template <class Etype>
void BSTree<Etype>::Insert(const Etype& insElem, TreeNode<Etype>* & TN)
{
    if (TN == NULL)
    {
        TN = new TreeNode<Etype>(insElem);
        size++;
    }
    else if (insElem < TN->element)
    {
        recentComp++; totalComp++;
        Insert(insElem, TN->left);
    }
    else if (insElem > TN->element)
    {
        recentComp++; totalComp++;
        Insert(insElem, TN->right);
    }
    // else insElem is in the tree already. We'll do nothing.
}
```

MUST PASS TN by reference!



Appendix E

Example Log Files

E.1 Spring 2005 Instructor Log File

The instructor log files contained the following strings representing key data used for log file processing and analysis **bold emphasis** represents actual string values in the log files as “keywords”

- **13:34:38.385** representing a TimeStamp (TS from now on) **Captured an Ink Stroke**
- **ID** of ink stroke (unique per page not document level)
- **Bounding Box** containing the ink stroke
- **Points** (some integer Z value representing the number of X, Y points captured): **Points (Z)**: followed by the Z number of points in the form **{X=coordinate (integer), Y=coordinate (integer)}**
- **Packet Descriptions**: (newline)
 - **Normal Pressure (Z)**: list of z integer values
 - **Timer Tick (Z)**: list of z integer associated (parallel) timer:pressure information
- Slide Navigation: TS **Current Slide changed from Slide 12 to Slide 11** - the pages need not be sequential. It is possible to have the form Slide X to Slide Y where X and Y are not in sequential order (i.e. Current Slide changed from Slide 10 to Slide 4)
- Tool usage
 - TS **Changed to Selection Tool** - no information on what was selected
 - TS **Changed to Stroke Eraser** - no information on what was erased
 - TS **Changed to Inking with Ink Pen**
 - TS **Changed to Inking with Highlighter**
 - TS **Changed to Laser Pointer**
 - TS **Current Slide changed from Slide X to Slide** - where X and Y are integer values
 - TS **Blank Slide Inserted after Slide X** - where X is an integer value
 - **Slide X Deleted** - where X is an integer value
- Closing application
 - TS **Presentation Saved**
 - TS **e-Fuzion Closed**
- Opening application - all log files have the following 4 lines unless an error occurred

- TS **e-Fuzion Loaded on a Tablet PC**
- TS **Changed to Inking with Ink Pen**
- TS **New Presentation Created**
- TS **Current Slide changed from Slide -1 to Slide 0**
- It is possible that a previous e-Fuzion file (efz) was loaded:
- TS **Presentation Opened** PATH - where PATH represents a path to an e-Fuzion file

14:02:28.546 New Presentation Created

14:02:28.676 Current Slide changed from Slide]1 to Slide 0

14:03:32.556 Captured an Ink Stroke

ID = 1

Bounding Box = {X=4517,Y=955,Width=464,Height=1299}

Points (26): Points (26): {X=4967,Y=1101} {X=4954,Y=1069} {X=4939,Y=1037} {X=4928,Y=1012}
 {X=4913,Y=986} {X=4895,Y=992} {X=4884,Y=1027} {X=4869,Y=1072} {X=4853,Y=1137} {X=4830,Y=1223}
 {X=4771,Y=1446} {X=4732,Y=1565} {X=4696,Y=1687} {X=4660,Y=1805} {X=4625,Y=1921} {X=4595,Y=2023}
 {X=4556,Y=2166} {X=4551,Y=2212} {X=4550,Y=2228} {X=4550,Y=2228} {X=4575,Y=2215} {X=4599,Y=2174}

Packet Descriptions:

Normal Pressure (26): 46 78 111 143 157 167 174 179 182 185 187 189 190 192 193 194 195 195

Timer Tick (26): 0 10 20 20 30 40 50 50 60 70 80 80 90 100 110 110 120 130 140 140 150 160 170

14:03:32.836 Captured an Ink Stroke

ID = 2

Bounding Box = {X=4729,Y=833,Width=726,Height=1387}

Points (25): Points (25): {X=4779,Y=846} {X=4759,Y=849} {X=4756,Y=862} {X=4756,Y=894}
 {X=4819,Y=1090} {X=4854,Y=1183} {X=4898,Y=1290} {X=4946,Y=1401} {X=4999,Y=1524} {X=5053,Y=1627}
 {X=5150,Y=1844} {X=5198,Y=1934} {X=5243,Y=2011} {X=5284,Y=2073} {X=5320,Y=2121} {X=5352,Y=2156}
 {X=5400,Y=2192} {X=5424,Y=2189} {X=5430,Y=2175} {X=5430,Y=2156} {X=5412,Y=2120}

Packet Descriptions:

Normal Pressure (25): 43 77 113 148 167 179 188 195 200 204 206 208 209 210 212 212 212 211

Timer Tick (25): 0 10 20 30 30 40 50 60 60 70 80 90 90 100 110 120 120 130 140 150 150 160 170

E.2 Spring 2007 Pilot Study

When the instructor “ends” the lecture the system automatically generated an XML file of all the interactions recorded during the class session.

- **UseInfo** contains an update of a user’s interaction with system and contains three attributes:
 - user the network identity of the student

- now the current system time of day in the form 2007-06-19 15:39:50Z
- status refers to either note-taking mode, outside the application, or performing assessment.
- **AssessmentTiming** takes regularly snapshots of current student status in completing any assessments in the system and contains the following attributes:
 - user is the net id of the student
 - objno is the objective the student is currently on
 - conf is the confidence level of a student’s submission
 - ass-type refers to either a pre-flight, post-flight, or assessment
 - time-type refers to start time or normal
 - time is a representation of the system’s current time of day in the form 2007-06-19 15:39:50Z
- **SessionInkInfo** contains the measurements of the following attributes:
 - totalink a measurement of the total number of ink strokes or ink density in some human readable form
 - totalpages total number of pages in the student’s note-taking whiteboard panel
 - specatorTime the amount of time spent ona page without inking in the form 00:01:13.7260128
 - inkTime the total time spent inking in the form 00:00:00
 - totalOtherToolUse number of times a user used tools other than notect=”3” spct=”15”¿
- **PageInkInfo** contains the measurements of page attributes including:
 - pageNo refers to the page number that the proceeding attributes in the list represent
 - strokeLength the total ink length on the page
 - strokeCount the total number of strokes on the page
 - avgStrokeSpeed the average speed of strokes on the page
 - deletions the numnber of strokes deleted on the page
 - spectactorTime the amount of time the page was displayed without any tablet interaction
 - inkingTime the amount of time spent inking on the page
 - objective the current objective the page is associated with

```
<?xml version="1.0"?>
<LogFile>
  <UserInfo user="superman" now="2007-06-19 15:39:16Z" status="Notetaking" />
  <AssesmentTiming user="scorpius" objno="0" assno="0" conf="100" ass-type="PreFlight"
    time-type="Start" time="2007-06-19 15:39:43Z" />
  <UserInfo user="scorpius" now="2007-06-19 15:39:40Z" status="Assesment" />
  <InkInfo user="superman" now="2007-06-19 15:40:15Z">
    <SessionInkInfo totalink="0" totalpages="1" specatorTime="00:01:13.7260128" inkTime="00:00:00"
      totalOtherToolUse="0" notect="3" spct="15">
        <PageInkInfo pageNo="0" strokeLength="0" strokeCount="0" avgStrokeSpped="0" deletions="0"
          spectactorTime="00:00:00" inkingTime="00:00:00" objective="0" />
      </SessionInkInfo>
    </InkInfo>
  </LogFile>
```

```

    </SessionInkInfo>
</InkInfo>
</LogFile>

<?xml version="1.0"?>
<LogFile>
<UseInfo user="jwilkis2" now="2007]06]19 15:39:16Z" status="Notetaking" />
<UseInfo user="peiper" now="2007]06]19 15:39:16Z" status="Notetaking" />
<UseInfo user="gjweber2" now="2007]06]19 15:39:17Z" status="Notetaking" />
<UseInfo user="bsulliv2" now="2007]06]19 15:39:17Z" status="Notetaking" />
<UseInfo user="akantor2" now="2007]06]19 15:39:17Z" status="Notetaking" />
<UseInfo user="rsclark2" now="2007]06]19 15:39:18Z" status="Notetaking" />
<UseInfo user="jwilkis2" now="2007]06]19 15:39:18Z" status="OutsideApplication" />
<UseInfo user="jwilkis2" now="2007]06]19 15:39:18Z" status="Notetaking" />
<UseInfo user="jwilkis2" now="2007]06]19 15:39:18Z" status="OutsideApplication" />
<UseInfo user="ehan5" now="2007]06]19 15:39:18Z" status="Notetaking" />
<UseInfo user="jwilkis2" now="2007]06]19 15:39:18Z" status="Notetaking" />
<UseInfo user="amduda" now="2007]06]19 15:39:21Z" status="Notetaking" />
<UseInfo user="apradow2" now="2007]06]19 15:39:26Z" status="Notetaking" />
<UseInfo user="gjweber2" now="2007]06]19 15:39:26Z" status="OutsideApplication" />
<UseInfo user="jwilkis2" now="2007]06]19 15:39:27Z" status="OutsideApplication" />
<UseInfo user="amorgan2" now="2007]06]19 15:39:27Z" status="Notetaking" />
<UseInfo user="jwilkis2" now="2007]06]19 15:39:29Z" status="Notetaking" />
<UseInfo user="evered" now="2007]06]19 15:39:31Z" status="Notetaking" />
<AssesmentTiming user="amorgan2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:39:43Z" />
<UseInfo user="amorgan2" now="2007]06]19 15:39:40Z" status="Assesment" />
<AssesmentTiming user="akantor2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:39:50Z" />
<UseInfo user="akantor2" now="2007-06-19 15:39:43Z" status="Assesment" />
<AssesmentTiming user="amduda" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:39:50Z" />
<UseInfo user="amduda" now="2007-06-19 15:39:43Z" status="Assesment" />
<AssesmentTiming user="apradow2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:40:04Z" />
<UseInfo user="apradow2" now="2007-06-19 15:39:57Z" status="Assesment" />

```

```

<AssessmentTiming user="rsclark2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:40:04Z" />
<UseInfo user="rsclark2" now="2007-06-19 15:39:58Z" status="Assesment" />
<AssessmentTiming user="ehan5" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:40:05Z" />
<UseInfo user="ehan5" now="2007-06-19 15:39:58Z" status="Assesment" />
<AssessmentTiming user="bsulliv2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:40:06Z" />
<UseInfo user="bsulliv2" now="2007-06-19 15:40:02Z" status="Assesment" />
<AssessmentTiming user="evered" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:41:01Z" />
<UseInfo user="evered" now="2007-06-19 15:40:02Z" status="Assesment" />
<UseInfo user="gjweber2" now="2007-06-19 15:40:04Z" status="Notetaking" />
<AssessmentTiming user="jwilkis2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06-19 15:40:11Z" />
<UseInfo user="jwilkis2" now="2007-06-19 15:40:05Z" status="Assesment" />
<AssessmentTiming user="gjweber2" objno="0" assno="0" conf="100" ass-type="PreFlight"
time-type="Start" time="2007-06]19 15:40:16Z" />
<UseInfo user="gjweber2" now="2007-06-19 15:40:09Z" status="Assesment" />
InkInfo user="jwilkis2" now="2007-06-19 15:40:15Z">
<SessionInkInfo totalink="0" totalpages="1" specatorTime="00:01:13.7260128" inkTime="00:00:00"
total0therToolUse="0" notect="3" spct="15">
<PageInkInfo pageNo="0" strokeLength="0" strokeCount="0" avgStrokeSpeed="0" deletions="0"
spectatorTime="00:00:00" inkingTime="00:00:00" objective="0" />
<PageInkInfo pageNo="1" strokeLength="0" strokeCount="0" avgStrokeSpeed="0" deletions="0"
spectatorTime="00:00:00.0156250" inkingTime="00:00:00" objective="0" />
<PageInkInfo pageNo="2" strokeLength="0" strokeCount="0" avgStrokeSpeed="0" deletions="0"
</SessionInkInfo>
</InkInfo>

```

References

- [1] G. D. Abowd. Classroom 2000: an experiment with the instrumentation of a living educational environment. *IBM Syst. J.*, 38(4):508–530, 1999.
- [2] D. Adams and M. Hamm. New directions in learning: Understanding media symbol systems. *Educational Technology*, 28(2):28–30, 1988.
- [3] C. Allen. The use of video in organizational studies. *ACM SIGCHI Bulletin*, 21(2):115–117, 1989.
- [4] C. Ames. Classrooms: Goals, structures, and student motivation. *Journal of Educational Psychology*, 84(3):261–71, 1992.
- [5] J. R. Anderson, C. F. Boyle, and B. J. Reiser. Intelligent tutoring systems. *Science*, 228(4698):456–462, 1985.
- [6] R. Anderson. Beyond powerpoint: Building a new classroom presenter, 2004.
- [7] R. J. Anderson, R. Anderson, T. Vandegrift, S. Wolfman, and K. E. N. Yasuhara. Promoting interaction in large classes with computer-mediated feedback. *Designing for Change in Networked Learning Environments*, pages 119–123, 2003.
- [8] Richard Anderson, Ruth Anderson, Crystal Hoyer, Craig Prince, Jonathan Su, Fred Videon, and Steven Wolfman. A study of diagrammatic ink in lecture. *Computers and Graphics*, 29(4):480–489, 2005.
- [9] J. D. Angers. *Integrating a Technology-Enriched Curriculum: Ethno-Case Study*. PhD thesis, 2004.
- [10] K. S. R. Anjaneyulu, R. A. Singer, and R. Harding. Usability studies of a remedial multimedia system. *Journal of Educational Multimedia and Hypermedia*, 1998.
- [11] Spoerri Anselm. Souvenir: flexible note-taking tool to pinpoint and share media in digital libraries, 2002.
- [12] J. Beavers, T. Chou, R. Hinrichs, C. Moffatt, M. Pahud, L. Powers, and J. Van Eaton. The learning experience project: Enabling collaborative learning with conferencexp. *Microsoft Research*, 2004.
- [13] H. J. Becker. Who’s wired and who’s not: Children’s access to and use of computer technology. *The Future of Children*, 10(2):44–75, 2000.
- [14] J. Beecher. Note-taking: What do we know about the benefits? 1988.
- [15] D. Berque, D. K. Johnson, and L. Jovanovic. Teaching theory of computation using pen-based computers and an electronic whiteboard. *Proceedings of the 6th annual conference on Innovation and technology in computer science education*, pages 169–172, 2001.
- [16] M. Berry and M. Hamilton. Mobile computing, visual diaries, learning and communication: changes to the communicative ecology of design students through mobile computing. *Proceedings of the 8th Australian conference on Computing education-Volume 52*, pages 35–44, 2006.

- [17] Simon Beth, Davis Krista, G. Griswold William, Kelly Michael, and Malani Roshni. Noteblogging: taking note taking public. *SIGCSE Bull.*, 40(1):417–421, 2008.
- [18] Simon Beth, Anderson Ruth, Hoyer Crystal, and Su Jonathan. Preliminary experiences with a tablet pc based system to support active learning in computer science courses, 2004.
- [19] Paul Black and Dylan Wiliam. Assessment and classroom learning. *Assessment in Education: Principles, Policy and Practice*, 5(1):7, 1998.
- [20] D. A. Bligh. Note taking in lectures. *What's the Use of Lectures?*, pages 129–147, 2000.
- [21] O. M. Bonastre, A. P. Benavent, and F. N. Belmonte. Pedagogical use of tablet pc for active and collaborative learning. In *International Professional Communication Conference, 2006 IEEE*, pages 214–218, 2006.
- [22] S. T. Bossert. Cooperative activities in the classroom. *Review of Research in Education*, 15:225–252, 1988.
- [23] P. Bristow. Unicast vs. multicast over wireless: a cross-disciplinary mindshare for educational application researchers. *Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education*, pages 242–244, 2006.
- [24] J. M. Carey. *Human Factors in Information Systems: An Organizational Perspective*. Intellect Books, 1991.
- [25] R. K. Carr and D. Shafer. *The Power of PenPoint*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1991.
- [26] V. Carter. Do media influence learning? revisiting the debate in the context of distance education. *Open Learning*, 11(1):31–40, 1996.
- [27] Montserrat Castelló and Carles Monereo. Students' note-taking as a knowledge-construction tool. *L1-Educational Studies in Language and Literature*, 5(3):265–285, 2005.
- [28] Z. Chambers, A. Chidanandan, R. Devasher, L. Merkle, M. Minster, S. Mitra-Kirtley, D. Mutchler, S. Sexton, A. Watt, J. Williams, and M. Zoetewey. What is beyond the laptop initiative? perhaps: Tablet pcs and dyknow vision software. In *Frontiers in Education Conference, 36th Annual*, pages 8–13, 2006.
- [29] L. Willis Cheryl and L. Miertschin. Tablet pc's as instructional tools or the pen is mightier than the 'board!, 2004.
- [30] Kevin Chevalier. *Interpretation of Box-and-Pointer Diagrams in Classroom Learning Partner*. PhD thesis, Massachusetts Institute of Technology, 2007.
- [31] E. H. H. Chi, P. Barry, J. Riedl, and J. Konstan. A spreadsheet approach to information visualization. *Information Visualization, 1997. Proceedings., IEEE Symposium on*, pages 17–24, 1997.
- [32] C. S. Chin and J. Tan. Use of sketchbook pro with tablet pc (tabsketch™) as a design thinking tool in the teaching and learning of design and technology. 2007.
- [33] Christine Chin. Classroom interaction in science: Teacher questioning and feedback to students' responses, 2006.
- [34] S. W. Choi and T. Tinkler. Evaluating comparability of paper-and-pencil and computer-based assessment in a k-12 setting. *annual meeting of the National Council on Measurement in Education, New Orleans, LA*, 2002.
- [35] R. Cicchino and D. Mirliss. Tablet pcs: A powerful teaching tool. *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2004*, pages 543–548, 2004.

- [36] B. R. Clark. *The Research foundations of graduate education: Germany, Britain, France, United States, Japan*. Berkeley: University of California Press, 1993.
- [37] S. Clark, L. Taylor, J. Pickering, and A. Wait. Understanding the impact of tablet pcs on students' learning and academics' teaching. In *Proceedings of the Second Innovation in Accounting and Corporate Governance Education Conference*, Hobart, Tasmania, 2007.
- [38] J. Cohen. Statistical power analysis. *Current Directions in Psychological Science*, 1(3):98–101, 1992.
- [39] K. P. Cross. Classroom research: Implementing the scholarship of teaching. *American Journal of Pharmaceutical Education*, 60(4):402–07, 1996.
- [40] M.S. Crowther, C.C. Keller, and G.L. Waddoups. Improving the quality and effectiveness of computer-mediated instruction through usability evaluations. *British Journal of Educational Technology*, 35(3):289–303, 2004.
- [41] L. Cuban. *Teachers and Machines: The Classroom Use of Technology Since 1920*. Teachers College Press, 1986.
- [42] L. Cuban. Computers meet classroom: Classroom wins. *The Teachers College Record*, 95(2):185–210, 1993.
- [43] L. Cuban. *How Teachers Taught: Constancy and Change in American Classrooms, 1890-1990*. Teachers College Press, 1993.
- [44] L. Cuban. *Oversold and Underused: Computers in the Classroom*. Harvard University Press, 2001.
- [45] S. Daniel, J. Ellis, and T. Truscott. Usenet, a general access unix network, 1980.
- [46] Berque Dave, Bonebright Terri, and Whitesell Michael. Using pen-based computers across the computer science curriculum, 2004.
- [47] Bargeron David and Moscovich Tomer. Reflowing digital ink annotations, 2003.
- [48] Lindquist David, Denning Tamara, Kelly Michael, Malani Roshni, G. Griswold William, and Simon Beth. Exploring the potential of mobile phones for active learning in the classroom, 2007.
- [49] A. de Vicente and H. Pain. Informing the detection of the students' motivational state: an empirical study. *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*, 2363:933–943, 2002.
- [50] A. de Vicente and H. Pain. Validating the detection of a student's motivational state. *Proceedings of the Second International Conference on Multimedia Information and Communication Technologies in Education (m-ICTE2003)*, pages 2004–2008, 2003.
- [51] T. del Soldato. Detecting and reacting to the learner's motivational state. *Proceedings of the Second International Conference on Intelligent Tutoring Systems*, pages 567–574, 1992.
- [52] T. Denning, W. G. Griswold, B. Simon, and M. Wilkerson. Multimodal communication in the classroom: what does it mean for us? *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 219–223, 2006.
- [53] J. M. Depew and D. C. Northway. Flat panel display with integrated electromagnetic pen digitizer, 2001.
- [54] L. P. Dringus. An iterative usability evaluation procedure for interactive online courses. *Journal of Interactive Instruction Development*, 7(4):10–14, 1995.
- [55] D. Ebert-May, C. Brewer, and S. Allred. Innovation in large lectures—teaching for active learning. *BioScience*, 47(9):601–607, 1997.

- [56] Saund Eric and Lank Edward. Stylus input and editing without prior selection of mode, 2003.
- [57] R. M. Felder. How students learn: adapting teaching styles to learning styles. pages 489–493, 1988.
- [58] D. Fisher, P. Cornwell, and J. Williams. Teaching dynamics using interactive tablet pc instruction software. In *Frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual*, pages S3J–3–S3J–4, 2007.
- [59] A. Gandolfo. Brave new world? the challenge of technology to time-honored pedagogies and traditional structures. *New Directions for Teaching and Learning*, 1998(76):23–38, 1998.
- [60] J. Gemmell, T. Montgomery, T. Speakman, and J. Crowcroft. The pgm reliable multicast protocol. *Network, IEEE*, 17(1):16–22, 2003.
- [61] Golovchinsky Gene and Denoue Laurent. Moving markup: repositioning freeform annotations, 2002.
- [62] Golovchinsky Gene, N. Price Morgan, and N. Schilit Bill. From reading to retrieval: freeform ink annotations as queries, 1999.
- [63] Winkenbach Georges and H. Salesin David. Computer-generated pen-and-ink illustration, 1994.
- [64] K. W. Gilcher and S. M. Johnstone. A critical review of the use of audiographic conferencing systems by selected educational institutions. 1988.
- [65] G. S. Goyins and M. F. Resman. Palm pressure rejection method and apparatus for touchscreens, 2001.
- [66] D. Abowd Gregory, G. Atkeson Christopher, Feinstein Ami, Hmelo Cindy, Kooper Rob, Long Sue, Sawhney Nitin, and Tani Mikiya. Teaching and learning as multimedia authoring: the classroom 2000 project, 1996.
- [67] D. Abowd Gregory, G. Atkeson Christopher, Brotherton Jason, Enqvist Tommy, Gulley Paul, and LeMon Johan. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting, 1998.
- [68] D. Abowd Gregory, Brotherton Jason, and Bhalodia Janak. Classroom 2000: a system for capturing and accessing multimedia classroom experiences, 1998.
- [69] J. Griffioen, W. B. Seales, and Jr. Lumpp, J. E. Teaching in realtime wireless classrooms. volume 2, pages 748–753 vol.2, 1998.
- [70] A. F. Hadwin, J. R. Kirby, and R. A. Woodhouse. Individual differences in notetaking, summarization, and learning from lectures. *Alberta Journal of Educational Research*, 45(1):1–17, 1999.
- [71] J. F. Hair Jr, R. E. Anderson, R. L. Tatham, and W. C. Black. *Multivariate data analysis: with readings*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1995.
- [72] M. J. Hannafin and S. M. Land. The foundations and assumptions of technology-enhanced student-centered learning environments. *Instructional Science*, 25(3):167–202, 1997.
- [73] James Hartley and Ivor K. Davies. Note-taking: A critical review. *Innovations in Education and Teaching International*, 15(3):207 – 224, 1978.
- [74] T. S. Hasselbring. Research on the effectiveness of computer-based instruction: A review. *International Review of Education*, 22:313–324, 1986.
- [75] D. C. A. Hillman. Learner-interface interaction in distance education: An extension of contemporary models and strategies for practitioners. *American Journal of Distance Education*, 8(2):30–42, 1994.

- [76] K. Hinckley, P. Baudisch, G. Ramos, and F. Guimbretiere. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 451–460, 2005.
- [77] K. Hinckley, G. Ramos, F. Guimbretiere, P. Baudisch, and M. Smith. Stitching: pen gestures that span multiple displays. *Proceedings of the working conference on Advanced visual interfaces*, pages 23–31, 2004.
- [78] U. Hoppe, A. Lingnau, I. Machado, A. Paiva, R. Prada, and F. Tewissen. Supporting collaborative activities in computer integrated classrooms—the nimis approach. *Proc. of CRIWG*, 2000.
- [79] L. G. Huettel, J. Forbes, L. Franzoni, R. Malkin, J. Nadeau, K. Nightingale, and G. A. Ybarra. Transcending the traditional: Using tablet pcs to enhance engineering and computer science instruction. In *Frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual*, pages S4J–1–T1A–6, 2007.
- [80] C. C. W. Hulls. Using a tablet pc for classroom instruction. pages T2G–1–T2G–6, 2005.
- [81] A. Iles, D. Glaser, M. Kam, and J. Canny. Learning via distributed dialogue: Livenotes and hand-held wireless technology. *Proceedings of Conference on Computer Support for Collaborative Learning, Boulder, CD*, 2002.
- [82] W. E. Jaber and D. M. Moore. A survey of factors which influence teachers’ use of computer-based technology. *International Journal of Instructional Media*, 26(3), 1999.
- [83] A. Pittman James. Handwriting recognition: Tablet pc text input. 40(9):49–54, 2007.
- [84] William G. Holliday Janice M. Bonner. How college science students engage in note-taking strategies. *Journal of Research in Science Teaching*, 43(8):786–818, 2006.
- [85] R. Jarrett and P. Su. *Building Tablet PC Applications*. 2003.
- [86] A. Brotherton Jason and D. Abowd Gregory. Lessons learned from eclass: Assessing automated capture and access in the classroom. *ACM Trans. Comput.-Hum. Interact.*, 11(2):121–155, 2004.
- [87] I. Hong Jason and A. Landay James. *SATIN: a toolkit for informal ink-based applications*. Proceedings of the 13th annual ACM symposium on User interface software and technology. ACM Press, San Diego, California, United States, 2000.
- [88] Brotherton Jason Alan. *Enriching everyday activities through the automated capture and access of live experiences. eclass: building, observing and understanding the impact of capture and access in an educational domain*. PhD thesis, 2001.
- [89] Roschelle Jeremy, Tatar Deborah, S. Raj Chaudhury, Dimitriadis Yannis, Patton Charles, and DiGiano Chris. Ink, improvisation, and interactive engagement: Learning with tablets. 40(9):42–48, 2007.
- [90] Lienhard Jochen and Lauer Tobias. *Multi-layer recording as a new concept of combining lecture recording and students’ handwritten notes*. Proceedings of the tenth ACM international conference on Multimedia. ACM Press, Juan-les-Pins, France, 2002.
- [91] P. M. Johnson. *Change in Classroom Practices of Technology Use by K-12 Teachers*. PhD thesis, 2006.
- [92] G. Tront Joseph. Facilitating pedagogical practices through a large-scale tablet pc deployment. 40(9):62–68, 2007.
- [93] M. Kadiyala and B. L. Crynes. Where’s the proof? a review of literature on effectiveness of information technology in education. volume 1, pages 33–37 vol.1, 1998.
- [94] S. Kagan. Dimensions of cooperative classroom structures. *Learning to cooperate, cooperating to learn*, pages 67–96, 1985.

- [95] C. Kasabach, C. Pacione, J. Stivoric, F. Gemperle, and D. Siewiorek. Digital ink: a familiar idea with technological might! *Conference on Human Factors in Computing Systems*, pages 175–176, 1998.
- [96] Kurihara Kazutaka, Goto Masataka, Ogata Jun, and Igarashi Takeo. Speech pen: predictive handwriting based on ambient multimodal recognition, 2006.
- [97] Mock Kenrick. Teaching with tablet pc’s. *J. Comput. Small Coll.*, 20(2):17–27, 2004.
- [98] R. G. Kern. Restructuring classroom interaction with networked computers: Effects on quantity and characteristics of language production. *Modern Language Journal*, 79(4):457–476, 1995.
- [99] N. Truong Khai and D. Abowd Gregory. *StuPad: integrating student notes with class lectures*. CHI ’99 extended abstracts on Human factors in computing systems. ACM Press, Pittsburgh, Pennsylvania, 1999.
- [100] K. A. Kiewra. Investigating notetaking and review: A depth of processing alternative. *Educational Psychologist*, 20(1):23–32, 1985.
- [101] K. A. Kiewra. Providing the instructor’s notes: An effective addition to student notetaking. *Educational Psychologist*, 20(1):33–39, 1985.
- [102] K. A. Kiewra. Notetaking and review: The research and its implications. *Instructional Science*, 16(3):233–249, 1987.
- [103] K. A. Kiewra. Providing study notes: Comparison of three types of notes for review. *Journal of Educational Psychology*, 80(4):595–97, 1988.
- [104] K. A. Kiewra. A review of note-taking: The encoding-storage paradigm and beyond. *Educational Psychology Review*, 1:147–172, 1989.
- [105] K. A. Kiewra. Note-taking functions and techniques. *Journal of Educational Psychology*, 83(2):240–45, 1991.
- [106] K. Koile. Improving learning in cs1 via tablet-pc-based in-class assessment. pages 119–126. ACM Press New York, NY, USA, 2006.
- [107] K. Koile, K. Chevalier, C. Low, S. Pal, A. Rogal, D. Singer, J. Sorensen, K. S. Tay, and K. Wu. Supporting pen-based classroom interaction: New findings and functionality for classroom learning partner. *To appear in Proceedings of First International Workshop on Pen-Based Learning Technologies*, 2007.
- [108] K. Koile, K. Chevalier, M. Rbeiz, A. Rogal, D. Singer, J. Sorensen, A. Smith, K. S. Tay, and K. Wu. Supporting feedback and assessment of digital ink answers to in-class exercises. *To appear in Proceedings of the Nineteenth Conference on Innovative Applications of AI*, 2007.
- [109] R.B. Kozma. Learning with media. *Review of Educational Research*, 61(2):179, 1991.
- [110] R.B. Kozma. Will media influence learning? reframing the debate. *Educational Technology Research and Development*, 42(2):7–19, 1994.
- [111] C. C. Kulik and J. A. Kulik. Effectiveness of computer-based instruction: An updated analysis. *Computers in Human Behavior*, 7:75–94, 1991.
- [112] J. A. Kulik. Effects of using instructional technology in elementary and secondary schools: What controlled evaluation studies say. *Arlington: sri International*, 2003.
- [113] R. Winer Laura and Cooperstock Jeremy. The intelligent classroom: changing teaching and learning with an evolving technological environment. *Comput. Educ.*, 38(1-3):253–266, 2002.
- [114] C. Liao, F. Guimbretière, R. Anderson, N. Linnell, C. Prince, and V. Razmov. Papercp: Exploring the integration of physical and digital affordances for active learning. *International Conference on Human-Computer Interaction (INTERACT), Sep*, 2007.

- [115] Chunyuan Liao, François Guimbretière, Ken Hinckley, and Jim Hollan. Papiercraft: A gesture-based command system for interactive paper. *ACM Transactions in Computer Human Interaction*, 14(4):1–27, 2008.
- [116] M. Lidström. *The Input Race: A comparative study between two input modalities*. Master’s thesis, Lulea University of Technology, 2007.
- [117] G. L. Lilly. *Correlating Technology Surveys and Third-and Fifth-Grade Proficiency Levels in Math and Reading Throughout Tennessee*. PhD thesis, East Tennessee State University, 2004.
- [118] Stifelman Lisa, Arons Barry, and Schmandt Chris. The audio notebook: paper and pen interaction with structured speech. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 182–189, 2001.
- [119] Jerome Eric Luczaj. A framework for e-learning technology. 2003.
- [120] D. Wilcox Lynn, N. Schilit Bill, and Sawhney Nitin. Dynamite: a dynamically organized ink and audio notebook. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 186–193, Atlanta, Georgia, United States, 1997. ACM Press.
- [121] M. E. E. Mac Yin. Notetaking-an overview, October 1991.
- [122] C. Malave, R. V. Belhot, and R. S. Figueiredo. New questions about learning styles. volume 3, pages S1D–S18 vol.3, 2003.
- [123] R. J. Marzano. *Designing a New Taxonomy of Educational Objectives*. Corwin Press, 2001.
- [124] R. J. Marzano and J. S. Kendall. The new taxonomy of educational objectives. *Corwin Press*, page 208, 2006.
- [125] R. J. Marzano, D. Pickering, and J. McTighe. Assessing student outcomes: Performance assessment using the dimensions of learning model, 1993.
- [126] Kam Matthew, Wang Jingtao, Iles Alastair, Tse Eric, Chiu Jane, Glaser Daniel, Tarshish Orna, and Canny John. Livenotes: a system for cooperative and augmented note-taking in lectures, 2005.
- [127] J. W. Maxwell. *Tracing the Dynabook: A Study of Technocultural Transformations*. PhD thesis, University of British Columbia, 2006.
- [128] A. Meyer. Pen computing: a technology overview and a vision. *ACM SIGCHI Bulletin*, 27(3):46–90, 1995.
- [129] S. Millinder. Positive impact of a tablet pc-based learning environment on student-perceived learning outcomes. In *Pen-Based Learning Technologies, 2007. PLT 2007. First International Workshop on*, pages 1–6, 2007.
- [130] S. M. Montgomery. Addressing diverse learning styles through the use of multimedia. volume 1, pages 3a2.13–3a2.21 vol.1, 1995.
- [131] M. G. Moore. Editorial: Three types of interaction. *The American Journal of Distance Education*, 3(2):1–6, 1989.
- [132] F. Mosteller. The ‘muddiest point in the lecture as a feedback device. *On Teaching and Learning: The Journal of the Harvard-Danforth Center*, 3:10–21, 1989.
- [133] B. A. Myers. A brief history of human-computer interaction technology. *interactions*, 5(2):44–54, 1998.
- [134] D. Norman. Organizational limits to hci. *Interview session at CHI*, 99, 1999.
- [135] C. Norris, T. Sullivan, J. Poirot, and E. Soloway. No access, no use, no impact: Snapshot surveys of educational technology in k-12. *Journal of Research on Technology in Education*, 36(1):15–28, 2003.

- [136] D. R. Olsen Jr, T. Taufer, and J. A. Fails. Screencrayons: annotating anything. pages 165–174. ACM Press New York, NY, USA, 2004.
- [137] Lourdes Ortega. Processes and outcomes in networked classroom interaction: Defining the research agenda for l2 computer-assisted classroom discussion. *Language Learning and Technology*, 1(1):82–93, 1997.
- [138] J. Ou, X. Chen, and J. Yang. Gesture recognition for remote collaborative physical tasks using tablet pcs. *Proc. of IX IEEE Intl. Conf. on Computer Vision, Work. on Multimedia Technologies in E-Learning and Collaboration (Nice, 2003)*, 2003.
- [139] R. Pargas, M. Cooper, C. Williams, and S. Bryfczynski. A tablet pc based interactivity tool for organic chemistry. *Pen-Based Learning Technologies, 2007. PLT 2007. First International Workshop on*, pages 6–6, 2007.
- [140] R. P. Pargas. Facilitating a course in tablet pc software development. *Proceedings of the 1st Workshop on the Impact of Pen-technology on Education, Purdue University, April*, pages 4–6, 2006.
- [141] R. P. Pargas and A. Goodbar. Frontiers in education conference, 36th annual. pages 14–15, 2006.
- [142] Chiu Patrick, Kapuskar Ashutosh, Reitmeier Sarah, and Wilcox Lynn. *NoteLook: taking notes in meetings with digital video and ink*. Proceedings of the seventh ACM international conference on Multimedia (Part 1). ACM Press, Orlando, Florida, United States, 1999.
- [143] C. Peiper, D. Warden, E. Chan, B. Capitanu, and S. Kamin. efuzion: development of a pervasive educational system. *ACM SIGCSE Bulletin*, 37(3):237–240, 2005.
- [144] Hardings Jens Perl. *Basis for a Methodology to Define, Validate and Apply Best Practices in a Computer-integrated Classroom*. PhD thesis, University of Chile, 2006.
- [145] J. Postel. User datagram protocol. *Aug*, 28:1–3, 1980.
- [146] Michel A. Rbeiz. *Semantic Representation of Digital Ink in the Classroom Learning Partner*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [147] Anderson Richard, Hoyer Crystal, Prince Craig, Su Jonathan, Videon Fred, and Wolfman Steve. *Speech, ink, and slides: the interaction of content channels*. Proceedings of the 12th annual ACM international conference on Multimedia. ACM Press, New York, NY, USA, 2004.
- [148] Anderson Richard, Anderson Ruth, Simon Beth, A. Wolfman Steven, VanDeGrift Tammy, and Yasuhara Ken. Experiences with a tablet pc based lecture presentation system in computer science courses, 2004.
- [149] Anderson Richard, Anderson Ruth, Davis Peter, Linnell Natalie, Prince Craig, Razmov Valentin, and Videon Fred. Classroom presenter: Enhancing interactive education with digital ink. 40(9):56–61, 2007.
- [150] C. Davis Richard, Lin James, A. Brotherton Jason, A. Landay James, N. Price Morgan, and N. Schilit Bill. A framework for sharing handwritten notes. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 119–120, San Francisco, California, United States, 1998. ACM Press.
- [151] M. Riel and K. Fulton. Technology in the classroom: Tools for doing things differently or doing different things. *annual meeting of the American Educational Research Association, San Diego, CA, April*, 1998.
- [152] C. Ringstaff and L. Kelley. The learning return on our educational technology investment. *A Review of Findings from Research*, pages 94107–1242, 2002.

- [153] Chung Ronald, Mirica Petrut, and Plimmer Beryl. Inkkit: a generic design tool for the tablet pc, 2005.
- [154] J. M. Roschelle, R. D. Pea, C. M. Hoadley, D. N. Gordin, and B. M. Means. Changing how and what children learn in school with computer-based technologies. *The Future of Children*, 10(2):76–101, 2000.
- [155] D. Rubine. Integrating gesture recognition and direct manipulation. *Proceedings of the Summer'91 USENIX Technical Conference*, pages 95–100, 1991.
- [156] Gunter Saunders and Fredrik Klemming. Integrating technology into a traditional learning environment: Reasons for and risks of success. *Active Learning in Higher Education*, 4(1):74–86, 2003.
- [157] L. Schomaker, E. de Leau, and L. Vuurpijl. Using pen-based outlines for object-based annotation and image-based queries. *Proceedings of the Third International Conference on Visual Information and Information Systems*, pages 585–592, 1999.
- [158] D. A. Schon. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, 1983.
- [159] R. Schwartz. The history of notes and domino. 2003.
- [160] Cynthia L. Selfe. Technology and literacy: A story about the perils of not paying attention. *College Composition and Communication*, 50(3):411–436, 1999.
- [161] B. Shackel. Human-computer interaction—whence and whither? *Journal of the American Society for Information Science*, 48(11):970–86, 1997.
- [162] A. Shesh and B. Chen. Smartpaper—an interactive and intuitive sketching system. *Accepted and to appear in Eurographics 2004*, 2004.
- [163] M. Shi and B. A. Bichelmeyer. Teachers’ experiences with computers: A comparative study. *JOURNAL OF EDUCATIONAL TECHNOLOGY AND SOCIETY*, 10(2):180, 2007.
- [164] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. *The Craft of Information Visualization: Readings and Reflections*, 2003.
- [165] Wang Shudong, Jiao Hong, Michael J. Young, Thomas Brooks, and John Olson. Comparability of computer-based and paper-and-pencil testing in k 12 reading assessments: A meta-analysis of testing mode effects. *Educational and Psychological Measurement*, 68(1):5–24, 2008.
- [166] L. J. Shuman, C. J. Atman, E. A. Eschenbach, D. Evans, R. M. Felder, P. K. Imbrie, J. McGourty, R. L. Miller, L. G. Richards, K. A. Smith, E. P. Soulsby, A. A. Waller, and C. F. Yokomoto. The future of engineering education. volume 1, pages T4A–1–T4A–15 vol.1, 2002.
- [167] Simeon J. Simoff and Mary Lou Maher. Analysing participation in collaborative design environments. *Design Studies*, 21(2):119–144, 2000.
- [168] B. Simon, R. Anderson, C. Hoyer, and J. Su. Preliminary experiences with a tablet pc based system to support active learning in computer science courses. volume 36, pages 213–217. ACM Press New York, NY, USA, 2004.
- [169] G. Singh, L. Denoue, and A. Das. Collaborative note taking. In *Wireless and Mobile Technologies in Education, 2004. Proceedings. The 2nd IEEE International Workshop on*, pages 163–167, 2004.
- [170] D. Sleeman and J. S. Brown. *Intelligent tutoring systems*. Academic Press, London; New York, 1982.
- [171] Amanda C. Smith. *Aggregation of Student Answers in a Classroom Setting*. PhD thesis, Massachusetts Institute of Technology, 2006.
- [172] J. Sneller. The tablet pc classroom: Erasing borders, stimulating activity, enhancing communication. In *Frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual*, pages S3J–5–S3J–10, 2007.

- [173] J. Soorma. Teacher concerns and attitudes during the adoption phase of one-to-one computing in early college high schools. 2008.
- [174] Y. Steinert and L. S. Snell. Interactive lecturing: strategies for increasing participation in large group presentations. *Medical Teacher*, 21(1):37–42, 1999.
- [175] Whittaker Steve, Hyland Patrick, and Wiley Myrtle. Filochat: handwritten notes provide access to recorded conversations. In *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*, pages 271–277, Boston, Massachusetts, United States, 1994. ACM Press.
- [176] J. Stifle. The plato iv architecture. 1971.
- [177] C. Stoll. *High-Tech Heretic: Why Computers Don't Belong in the Classroom and Other Reflections by a Computer Contrarian*. Random House Inc. New York, NY, USA, 1999.
- [178] Kah Seng Tay. *Improving Digital Ink Interpretation through Expected Type Prediction and Dynamic Dispatch*. PhD thesis, Massachusetts Institute of Technology, 2008.
- [179] D. Tenneson and S. Becker. Chempad: generating 3d molecules from 2d sketches. *International Conference on Computer Graphics and Interactive Techniques*, 2005.
- [180] M. D. Theys, K. Lawless, and S. George. Tablet computers and the traditional lecture. In *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*, pages T2G–7–12, 2005.
- [181] C. Thompson. Powerpoint makes you dumb. *TECHNOLOGY*, 2003.
- [182] S. Butler Timothy. Clearpen: improving the legibility of handwriting. In *CHI '03 extended abstracts on Human factors in computing systems*, pages 678–679, Ft. Lauderdale, Florida, USA, 2003. ACM Press.
- [183] C. Track. The nsf national science, technology, engineering, and mathematics education digital library (nsdl) program. *D-Lib Magazine*, 8(11):1082–9873, 2002.
- [184] J. Tront, G. Scales, D. Olsen, and J. Prey. Deploying tablet pcs in an engineering education environment. In *Frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual*, pages T3G–24–T3G–29, 2007.
- [185] J. G. Tront, V. Eligeti, and J. Prey. Classroom presentations using tablet pcs and writeon. *Proc. 2006 Frontiers in Education Conference. San Diego, CA: IEEE*, 2006.
- [186] E. Tufte. Powerpoint is evil. *Wired Magazine, Septembre*, 2003.
- [187] E. R. Tufte. *The visual display of quantitative information*. Graphics Press Cheshire, CT, USA, 1986.
- [188] E. R. Tufte. *Envisioning information*. Cheshire, Conn.: Graphics Press, 1990.
- [189] E. R. Tufte and J. R. M. Reviewer. Visual explanations: Images and quantities, evidence and narrative. *Computers in Physics*, 12:146, 1998.
- [190] Razmov Valentin and Anderson Richard. Pedagogical techniques supported by the use of student devices in teaching software engineering, 2006.
- [191] T. Volery and D. Lord. Critical success factors in online education. *International Journal of Educational Management*, 14(5):216–223, 2000.
- [192] E. D. Wagner. In support of a functional definition of interaction. *American Journal of Distance Education*, 8(2):6–29, 1994.
- [193] E. J. Webb, D. T. Campbell, R. D. Schwartz, and L. Sechrest. *Unobtrusive Measures (Revised Edition)*. Thousand Oaks, CA: Sage Publications, 2000.

- [194] H. Wenglinsky. *Using technology wisely: the keys to success in schools*. New York: Teachers College Press, 2005.
- [195] M. Wilkerson, W. G. Griswold, and B. Simon. Ubiquitous presenter: increasing student access and control in a digital lecturing environment. *Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 116–120, 2005.
- [196] J. R. Williams, A. H. Slocum, and A. Sanchez. Advanced technology for learning, vol. 2, no. 1, 2005 inkboard-tablet pc enabled design-oriented learning h. ning. 2005.
- [197] K. Williams. *Beliefs about technology integration support factors held by school leadership and school faculty: A mixed methods study*. PhD thesis, 2006.
- [198] R. L. Williams and A. C. Eggert. Notetaking in college classes: Student patterns and instructional strategies. *JGE: The Journal of General Education*, 51(3), 2002.
- [199] M. Windschitl. The www and classroom research: What path should we take. *Educational Researcher*, 27(1):28–33, 1998.
- [200] J. C. Wise, R. Toto, and Lim Kyu Yon. Introducing tablet pcs: Initial results from the classroom. In *Frontiers in Education Conference, 36th Annual*, pages 17–20, 2006.
- [201] D. Wooley. Plato: The emergence of online community. *On-line version at <http://www.thinkofit.com/plato/dwplato.htm>*, 1994.
- [202] D. Wortman and P. Rheingans. Visualizing trends in student performance across computer science courses. *Technical Symposium on Computer Science Education*, pages 430–434, 2007.
- [203] Kenneth D Wu. *Interpretation and Aggregation of Marks in Classroom Learning Partner*. PhD thesis, Massachusetts Institute of Technology, 2008.
- [204] Li Yang, Hinckley Ken, Guan Zhiwei, and A. Landay James. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proceeding of the SIGCHI conference on Human factors in computing systems*, pages 461–470, Portland, Oregon, USA, 2005. ACM Press.
- [205] J. R. Young. When good technology means bad teaching. *Chronicle of Higher Education*, 51(12):2, 2004.

Curriculum Vitae

Chad E. Peiper

2513 Hallbeck Dr.
Champaign, IL,
61822

Phone 217-202-5764
E-mail
peiper@cs.uiuc.edu

Education

- 2002–2008 **University of Illinois at Urbana-Champaign**
Ph.D. in Computer Science
Dissertation-Teacher's Dashboard: Active and Passive Monitoring of the Classroom Using Tablet PCs.
- 1998–2000 **University of Illinois at Urbana-Champaign**
M.C.S. in Computer Science
- 1994–1996 **University of California, Santa Cruz**
M.A. in Musicology - emphasis in Computer-Based Research
Thesis published as "Dissonance and Genre in Corelli's Trio Sonatas: A LISP-Based Study of Opp. 1 and 2,," Computing in Musicology, 10 (1995-96), 34-48.
- 1990–1994 **Case Western Reserve University - Cleveland, Ohio**
B.S. Music Education, *Magna cum Laude*

Teaching Experience

- 2000–2007 **Visiting Lecturer**, University of Illinois at Urbana-Champaign
CS400: Data Structures for Non-CS Majors
CS411: Introduction to Databases
CS302: Software Design and Development
CS225: Data Structures and Software Principles
CS125: Introduction to Computer Science
- 1998–2000 **Head Teaching Assistant**, University of Illinois at Urbana-Champaign
CS225: Data Structures and Software Principles
CS223: An introduction to object-oriented programming in C++
- 1997 **Violin and Orchestral Instructor**, Illinois Summer Youth Music, University of Illinois at Urbana-Champaign
ISYM is a comprehensive and intensive program of music instruction for student musicians with objectives to develop musical skills, and to improve their understanding and appreciation of music.

- 1995–1996 **Summer Teaching Assistant**, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University
 – Computer Assisted Musicology
 A seminar which provided a comprehensive introduction to computer-assisted research using the Humdrum Toolkit. This software package manipulates computer-based scores, tabulators, and other documents in order to solve a wide variety of analytic problems (UNIX operating system).
 – Introduction to Algorithmic Composition: covered topics such as data representation, techniques employing random selection, algorithmic editing, pattern generation and scheduling, sound synthesis as used in MIDI, Music Kit, Common Lisp Music, and Common Music Notation (LISP programming language).
- 1994–1996 **Teaching Assistant**, University of California at Santa Cruz
 Conducted bi-weekly labs in upper division theory, keyboard, and musicianship.
- 1994 **Student Teacher**, Heights High School, Cleveland, OH
 Instructed advanced placement music theory and analyses courses.

Research Experience

- 2004–2008 **Director of the SLICE Educational Laboratory** Responsibilities include system maintenance, programming, systems analysis and design, research and development of educational software, and training to faculty.
- 2000–2004 **e-Fuzion Solution Developer** Over the course of four years, we have investigated and experimented with ways to enhance learning of our students. This research explored the potential of wireless devices and associated collaborative software developed for the improvement of learning by undergraduate students in a large class setting.
- 1999–2000 **Research Assistant**, Beckman Institute, University of Illinois, Urbana-Champaign
 Project e-violin : A Violin Controller for Real-Time Audio
 Synthesis. The e-violin is an electric violin whose position, orientation, pitch, and loudness are measured continuously and used to compute sounds in real time with NCSA's VSS software. It offers visual feedback through an on-bridge display or the CAVE VR theater.
- 1996–1999 **Research Assistant**, University of Illinois, Urbana-Champaign
 Database Administrator (DBA) for the Hymn Tune Index. Responsibilities included query optimization of the database and the design and of a user interface to interact with the HTI Corel Paradox database through the World Wide Web (<http://hymntune.library.uiuc.edu/default.asp>).
- 1992–1994 Music Scores Application Project, Case Western Reserve
 Joint project with CWRU and IBM Corporation Interactive linking of scores, analyses, and recordings.

Presentations

- 2008 Illinois Council of Teachers of Mathematics, 59th Annual Meeting: Connections in the Mathematics Landscape—A Dashboard for Teaching: New Classroom Technology, October 16-18, 2008.
- 2008 Faculty Summer Institute 2008—Pen-Based Tablet Computing Using “Rich Ink” for the Classroom.
- 2008 Center for Teaching Excellence Faculty Retreat: Teacher’s Dashboard - UIUC
- 2005 Naperville Community Unit School District 203—Tablet PCs in K-12 Learning Environments
- 2004 The 2004 National Summit for Community Wireless Networks—Building Wireless Applications
- 2003 E-Learn Conference: Validation of e-Fuzion: a Wireless Classroom Technology
- 2002 JETT Conference—How Mighty is the Digital Pen?
- 2000 TeachIT Conference—e-Fuzion Demonstration

Publications

- 2008 C. Peiper, S.N. Kamin. A Non-Traditional Class Structure For Introductory Programming, Using Tablet PC's. Submitted for publication.
- 2008 S.N. Kamin, M. Twidale, B. Capitanu, C. Peiper. A 'Teacher's Dashboard' For A High School Algebra Class. To appear, Workshop on Impact of Pen-enabled Technology in Education (WIPTE '08), Purdue Univ. Oct. 2008.
- 2008 A System for Developing Tablet PC Applications for Education. Sam Kamin, Michael Hines, Chad Peiper, Boris Capitanu. Special Interest Group on Computer Science Education (SIGCSE 2008). March 12-15, 2008. Portland, Oregon.
- 2007 T. L. Wentling, J. Park, C. Peiper (2007). Learning gains associated with annotation and communication software designed for large undergraduate classes *Journal of Computer Assisted Learning* 23 (1), 36-46.
- 2005 Peiper, C., Warden, D., Chan, E., Capitanu, B., and Kamin, S. 2005. eFuzion: development of a pervasive educational system. In *Proceedings of the 10th Annual SIGCSE Conference on innovation and Technology in Computer Science Education* (Caparica, Portugal, June 27 - 29, 2005). ITiCSE '05. ACM Press, New York, NY, 237-240.
- 2005 Chad Peiper, David Warden, Ellick Chan, Roy H. Campbell, Sam Kamin, Tim L. Wentling: Applying Active Space Principles to Active Classrooms. *PerCom Workshops 2005*: 97-102.
- 2004 Chad Peiper, Ellick Chan, Roy Campbell, Jim Bresler, Jalal Al-Muhtad: Expanding Education through Active Space Collaboration. *PerCom Workshops 2004*: 236-240.
- 2003 Wentling, T.L., Park, J. H., Peiper, C., & Wadsworth, A. Facilitating Learning in a Wireless Classroom. *Academy of Human Resource Development (AHRD) Conference*. Minneapolis, MN.
- 2003 Tim Wentling, C. Peiper, P. Bristow, and J.H. Park, "Validation of e-Fuzion: a Wireless Classroom Technology," *Proc. World Conference on E-Learning in Corp., Govt., Health & Higher Ed.*, Vol. 2003, Issue. 1, 2003, pp. 343-350.
- 2003 Chad Peiper, David Warden, Guy Garnett: An Interface for Real-time Classification of Articulations Produced by Violin Bowing. *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, Montreal, Canada.
- 1996 Chad Peiper, "Dissonance and Genre in Corelli's Trio Sonatas: A LISP-Based Study of Opp. 1 and 2," *Computing in Musicology*, 10 (1995-96), 34-48.

Technical Skills

CMS Agora, DotNetNuke, Moodle, Microsoft SharePoint Server, Sakai

Databases Access, SQL Server, MySQL, and Oracle

Languages C, C++, C#, Lisp, Java, L^AT_EX, Perl, PHP, and Python

OS Mac-OS, Linux, Windows and UNIX

Dev. Tools Visual Studio, Eclipse, AccuRev, Caliber, SourceSafe, CVS, SVN, JIRA, Mantis, Bugzilla

Awards and Honors

2000 Advanced Graduate Teaching Certificate, University of Illinois at Urbana-Champaign

2000 Incomplete List of Teachers Ranked as Excellent by Their Students—CS225 and CS110C

1999 Theodore Presser Graduate Music Award, University of Illinois at Urbana-Champaign

1999 Graduate Teaching Certificate, University of Illinois at Urbana-Champaign

1999 Teaching Assistant of the Year Award, Department of Computer Science, University of Illinois at Urbana-Champaign

1999 Incomplete List of Teachers Ranked as Excellent by Their Students—CS223

1998 *AE*³ - The Academy For Excellence In Engineering Education, University of Illinois at Urbana-Champaign

1990–1994 Dean's Highest Honors, Case Western Reserve University

1994 Mortar Board, National Honor Society for Leadership, Service, and Academic Merit

1994 Mu Phi Epsilon, National Honor Society for Excellence in Service and Education

1993 Max and Eva Apple Scholarship, Leadership, and Service, Case Western Reserve University